(2)

FS-87-193
SEPTEMBER 1987

INFRARED SHIP DETECTION AT LOW SIGNAL-TO-NOISE

FLEET SYSTEMS DEPARTMENT
THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
FS-87-193

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
The Johns Hopkins University
Applied Physics Laboratory

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Johns Hopkins Road
Laurel, MD 20707

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
Infrared Ship Detection at Low Signal-to-Noise

12. PERSONAL AUTHOR(S)
R. A. Steinberg and J. J. Rivera

13a. TYPE OF REPORT
Interim

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
1987 September

15. PAGE COUNT
81

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

A multi-resolution signal processing algorithm ("MRSI") was devised to maximize the detection range of imaging infrared seekers against ship targets. Image-based simulations performed on an IBM PC AT verify that the new algorithm provides a signal-to-noise enhancement (relative to hot-spot detection) given by $\eta\sqrt{N_{PIX}}$, where N_{PIX} is the number of pixels occupied by the target in the image, and η=0.81. The processor efficiency is nearly invariant with respect to sensor/ship range (i.e.,scale), viewing aspect, and ship class. Probabilities of detection, determined via 2500 image-based Monte Carlo simulations, agree with well-known analogous results of psychophysical experiments performed with human subjects. Regardless of whether final target classification is performed on-board the missile via a computer algorithm, or off-board by a person inspecting imagery telemetered from the missile, MRSI substantially improves predicted system performance.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL
Richard Steinberg

22b. TELEPHONE *(Include Area Code)*
(301) 953-6527

22c. OFFICE SYMBOL
Group F1F

83 APR edition may be used until exhausted.
All other editions are obsolete.

FS-87-193
SEPTEMBER 1987

INFRARED SHIP DETECTION AT LOW SIGNAL-TO-NOISE

By: R. A. Steinberg
J. J. Rivera

FLEET SYSTEMS DEPARTMENT
THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY
Johns Hopkins Road, Laurel, Maryland 20707
Operating Under Contract N00039-87-C-5301 with the Department of the Navy

ABSTRACT

A multi-resolution signal processing algorithm ("MRSI") was devised to maximize the detection range of imaging infrared seekers against ship targets. Image-based simulations performed on an IBM PC AT verify that the new algorithm provides a signal-to-noise enhancement (relative to hot-spot detection) given by $\eta N_{PIX}^{1/2}$, where N_{PIX} is the number of pixels occupied by the target in the image, and $\eta \cong 0.8$. The processor efficiency, η, is remarkably invariant with respect to sensor/ship range (i.e., scale), viewing aspect, and ship class. Probabilities of detection, determined via 2500 image-based Monte Carlo simulations, agree with well-known analogous results of psychophysical experiments performed with human subjects. A new method is presented for supressing false alarms caused by cloud reflections from the sea surface. Regardless of whether final target classification is performed on-board the missile by a computer algorithm or off-board by a person inspecting imagery telemetered from the missile, MRSI substantially improves predicted system performance.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLFS

1.0 INTRODUCTION AND SUMMARY

Self-guided missiles have on a number of recent occasions demonstrated high effectiveness in operations against surface ships [1]. Both infrared (IR) and radio frequency (RF) means have historically each found application as the sensing element in antiship missile (ASM) guidance units. Some well-known advantages of IR technology in this application include [2]

- passive operation,
- good resistance to jamming, and
- high spatial resolution.

The impetus for high spatial resolution follows from potential operational needs for target classification: assuming adequate signal to-noise ratio (SNR), high classification accuracy requires high spatial resolution, regardless of whether the imagery is interpreted by a man[3,4] or processed by a computer [5]. Perhaps the key limitation of IR ASM seekers is limited range performance under conditions of poor atmospheric visibility. Thus, the purpose of the signal processor described in this report (referred to as "MRSI" for Multi-Resolution Spatial Integrator) is to perform detection of targets having minimum contrast relative to their background, in a digitized two-dimensional image. The MRSI processing approach could be used to optimize the detection range of infrared sensors against large area targets such as ships.

Means previously investigated for maximizing IR sensor SNR, relevant to the ASM seeker application, include waveband optimization [6,7], advanced IR detector developments [8,9], and multi-frame image processing [10]. Nonetheless, acquisition range remains an important performance index, and improvements in acquisition range an important objective, in the design of next generation antiship seekers. The single-frame signal processing approach discussed in this report is complementary to earlier approaches for SNR optimization.

The idea for our new signal processing concept was anticipated by a brief study indicating that detection ranges obtained by human observers of visual displays could under a broad variety of conditions greatly exceed ranges obtained by a hot-spot detection algorithm [11]. (This analysis was based on IR ship signatures cataloged in [12] and the human observer/display model provided in [3,4].) Thus, the predicted performance of an archetypal "human observer" became for us the standard against which to guage the performance of proposed ship detection algorithms. A similar outlook was adopted in [13] for assessing the performance of a digitally-implemented autotracker algorithm.

Our proposed solution to the ship detection problem is comprised of a separable multi-resolution filter, which we refer to as a Multi-Resolution Spatial Integrator (MRSI), first described in [14].

An early multi-resolution processing application involved the use of quad trees for encoding 1-bit images [15]. A variety of additional multi-resolution image processing applications and computational methods and costs are discussed in [16,17]. The multi-resolution processor discussed in this report is highly specialized to maximize SNR against rectangular and nearly-rectangular targets of known orientation, but whose size, aspect ratio, and position in the image are unknown. The MRSI approach is by design ideally suited for processing data from a linear array scanning sensor, i.e., image data acquired sequentially by column.

The time domain responses and SNR enhancement provided by MRSI processing are amenable to exact analytical computation for only a few geometrically simple target shapes. A number of such analytically predictable processor responses was used to validate a general image-based computer simulation. Test targets used for this purpose included point targets, bar targets, white gaussian noise, and bar targets in noise. The computer simulation was then exercised against a variety of ship images representative principally of frigates and aircraft carriers, seen from ranges of from 5 to 30 nautical miles (nmi) at viewing aspects from 90° (beam) to 10°. Threshold SNRs for detection were established in simulation by adding to each test image progressively increasing levels of noise.

It was found that MRSI processing, under a broad variety of conditions, provides about ten-fold SNR enhancement ("processing gain") relative to hot-spot detection. Moreover, the processing gain turns out to have been computable, with no more than about 10% maximum error in every case, as

$$G = \eta N_{PIX}^{1/2} \qquad (1)$$

where N_{PIX} is the number of pixels occupied by the target in the image, and η is given by

$$\eta = 0.81. \qquad (2)$$

The processing constant, η, is remarkably invariant with respect to sensor/ship range (i.e., scale), viewing aspect, and ship type.

The existance of a simply-computed analytical expression for processing gain, Eq. 1, is in itself a significant result, as it eliminates the need to perform image-based simulations in connection with cost/performance trade studies involving such additional aspects of design as focal plane sensitivity requirements. Also, an expression for η has been derived that can be evaluated simply from ship profiles, such as those provided in Jane's Fighting Ships [18]. Consequently, our results are readily generalized to ship types other than those for which detailed simulations have been conducted. At least as regards first-order system performance assessments, Eq. 1 may be used as a vastly simpler substitute for detailed image-based simulations.

The principal simulation results are plots of detection probability (P_D) vs. SNR (cf. Figs. 42 and 43), based on 2500 image-based Monte Carlo trials. A curve fit to the numerically-developed probabilities of detection appears to provide a universal curve that can be used predictively for ships of differing class, observed at a variety of aspects and distances. Interestingly, the curve fit to MRSI/Monte Carlo P_D results coincides with a previously published curve fit to data obtained from visual detection experiments performed with human observers of TV displays [3].

Two ASM system concepts are discussed in Section 15.0, with each concept employing a different means of target classification, viz., classification performed on-board the missile by an autonomous target recognition (ATR) algorithm or off-board by a person inspecting imagery telemetered from the missile. Using MRSI to cue a slow scan/high sensitivity "classification mode", as proposed by W. J. Tropf (APL), results in substantial improvements in predicted performance for both system concepts.

In Section 16.0 we present a new method for suppressing false alarms caused by cloud reflections from the sea surface.

The image-based simulation itself is coded in FORTRAN and operates on ship images of 64 x 256 pixels, at 35 spatial resolutions (App. A). The program runs both on the APL National Advanced Systems (NAS) mainframe and on an IBM PC AT computer, and requires about 450K bytes of memory. All simulation results provided in this report (more than 2800 processed images) were developed on an IBM PC AT.

2.0 ASSUMPTIONS AND DEFINITIONS

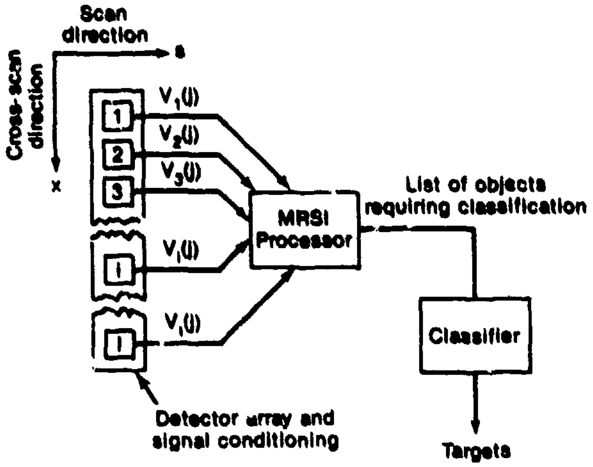


Figure 1. Scanning IR seeker concept. Vertical linear array of I infrared detectors is scanned in azimuth, generating I digital waveforms which are then input to the MRSI processor. MRSI's output is a list of objects, each characterized in terms of its position, dimensions (az x el), and brightness.

It is assumed that the MRSI signal processor is attached to an imaging sensor that performs surveillance of a scene once very T_F seconds, where the interval T_F is called the sensor's "frame time" (Fig. 1).

The angular dimensions of a single detector's field-of-view, denoted as α (in-scan) and β (cross-scan) are collectively referred to as the seeker's "spatial resolution" or "instantaneous field-of-view" (IFOV) and are specified in milliradian (mr) units. The voltage, $V_i(j)$, read out from detector i at sampling time j is presumably proportional to the integral of the scene radiance over a rectangular region of angular dimensions $\alpha \times \beta$.

The viewed scene is assumed to be a rectangle of angular dimensions A × B, mr. The sensor IFOV subdivides the total scene into a rectangular grid of samples, each of angular dimensions $\alpha \times \beta$, mr. The total scene (also called a "frame") and the sampling geometry are depicted in Fig. 2. Also shown in Fig. 2 is a rectangular coordinate frame (s,x). Samnles are obtained on a uniform rectangular grid, with adjacent samples separated by angle α/u in the s-direction and by angle β/v in the x-direction. (β/v is, of course, the angular separation of adjacent detector elements in the detector array.) As mentioned above, each sample is considered to be an integral of the scene brightness over a small rectangular region of angular dimensions $\alpha \times \beta$. As shown in Fig. 2, dimensions (A, α, α/u) are parallel to the s-axis, while dimensions (B, β, β/v) are parallel to the x-axis.

Depending on how the sensor is implemented, the spatial samples may be gapped, overlapped, or precisely adjacent. Figure 2 depicts an example for which the samples are gapped in both the s- and x-directions, i.e., for which

$$u, v < 1. \quad (3)$$

Samples are precisely adjacent in the s-direction when

$$u = 1. \quad (4)$$

Samples are overlapped in the s-direction when

$$u > 1. \quad (5)$$

Analogous to Eqs. 4 and 5, samples are precisely adjacent and overlapped in the x-direction when $v=1$ and $v>1$, respectively.

In general, the ratios of sample size to sample spacing, u and v, are different in value. For example, depen-

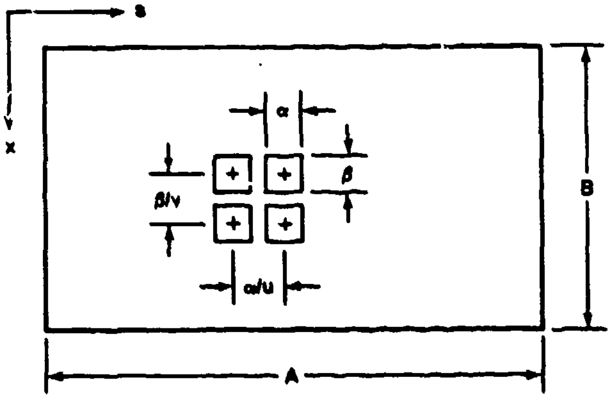


Figure 2. Total scene is a rectangle of angular dimensions, AxB. Spatial samples are of angular dimensions, $\alpha \times \beta$. Samples are obtained on centers displaced by angles α/u and β/v. All angles are measured in milliradians (mr.)

ding on the sensor's implementation, samples may be adjacent in s and gapped in $x(u=1, \upsilon<1)$ or overlapped in s and adjacent in $x(u>1, \upsilon=1)$; etc.

In the following discussion, a scanning sensor implementation is assumed, in which the s-coordinate (Fig. 2) coincides with the direction of scan, and the x-coordinate is variously referred to as the "cross-scan" or "cross-channel" coordinate. Quantity u is conventionally referred to as the in-scan sample factor, having units of samples per dwell.

Although the following discussion adopts the parlance of scanning sensors, the MRSI processor is readily adapted to staring mosaic sensors as well.

Regardless of the numerical values of the parameters defined in Fig. 2, the sensor develops every T_F seconds a matrix of numbers, $\{V_i(j)\}$, in which each matrix element is proportional to the brightness of the scene, averaged over a neighborhood of dimensions $\alpha \times \beta$, centered on the angular coordinates $(j\alpha/u, i\beta/\upsilon)$.

The detector index, i, is also the cross-channel image coordinate, taking on the range of values

$$1 \le i \le I = [B\upsilon/\beta], \tag{6}$$

where square brackets $[\cdot]$ denote the integer part of their argument. Similarly, the discrete time variable, j, takes on the range of values

$$1 \le j \le J = [Au/\alpha]. \tag{7}$$

During any given sampling interval (i.e., for a given value of j), the MRSI processor's input is a vector V_i of dimension I. Equivalently, the processor operates upon an input consisting of I digital waveforms.

If the IR seeker were positioned in the far field of a bar target, the voltage output of each detector channel would appear approximately as shown in Fig. 3.

For the purposes of this discussion, an object of angular dimensions:

γ_s, in the direction of scan ("in-scan")
γ_x, cross scan,

is said to be "resolved in-scan" when $\gamma_s > \alpha$, and "resolved cross-scan" when $\gamma_x > \beta$. An object is said to be "fully resolved" when it is resolved both in-scan and cross-scan.

As discussed in [11], ship targets are fully resolved at initial detection, for all cases of practical interest.

The "channel signal-to-noise ratio", SNR_C, is calculated as (cf. Fig. 3b)

$$SNR_C = (V_h - V_c)/V_{rms} \tag{8}$$

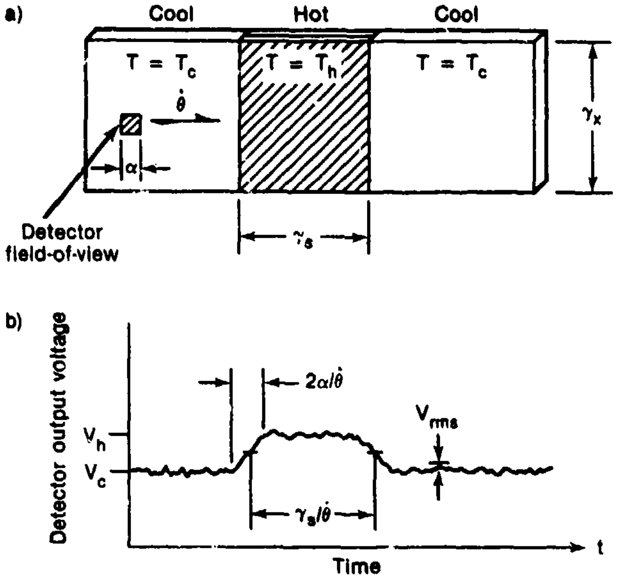


Figure 3. Part a): The field-of-view of an IR detector is scanned across an ideal bar target. T denotes temperature. Detector and target dimensions, α and γ respectively, are in angle units (mrad). Angular rate of scan is a constant, $\dot{\theta}$, mrad/sec.
Part b): Appearance of the voltage waveform obtained at the detector's output for the situation depicted in part a).

i.e., as the ratio of peak signal to root-mean-square (RMS) noise, at the output of a single detector channel.

A basic measure of seeker sensitivity is its "noise equivalent temperature", NEΔT, calculated as (cf. Fig. 3)

$$NE\Delta T = (T_h - T_c)/SNR_C \tag{9}$$

with SNR_C given by Eq. 8.

The sensitivity parameter NEΔT may be calculated using measurements made against calibrated laboratory targets, using Eqs. 8 and 9. Quantity NEΔT may also be calculated (or predicted prior to sensor fabrication) as a function of such key seeker optical characteristics as aperture, $\dot{\theta}$, α, detector detectivity (D*), etc. [19].

The number of pixels (spatial resolution elements) occupied by a rectangular target of size $\{\gamma_s$ (in-scan) $\times\ \gamma_x$ (cross-scan)$\}$ is given by

$$N_{PIX} = \gamma_s\gamma_x/\alpha\beta \tag{10}$$

Equation 9 is presented above from the perspective of using calibrated bar target data to obtain a measured va!ue of NEΔT, as a means of characterizing seeker sensitivity. Once NEΔT has been obtained, a variant of Eq. 9 may then be used to predict the SNR expected in a tactical scenario, viz.,

$$SNR_C = \Delta T/NE\Delta T \qquad (11)$$

where the quantity ΔT is now the "apparent" contrast temperature of the tactical target. Apparent ship contrast temperature, ΔT, is sensitive to a wide variety of parameters characteristic of the target, the background, and the atmosphere [20].

Quantities ΔT and SNR_C are most generally determined as spatial distributions since they may take on different values for each pixel in the frame.

3.0 HUMAN VISION AS A MODEL OF PERFECT PROCESSING

At least as regards spatial resolution (α, β), requirements for target recognition are at odds with requirements for long range acquisition. Assuming adequate SNR, high accuracy in target recognition requires high spatial resolution, regardless of whether the imagery is interpreted by a man [3,4] or processed by a computer [5]. Thus, the objectives of recognition are furthered by making the number of pixels on target (N_{PIX}) very large. On the other hand, SNR is maximized when the resolution element is matched in size to the target, i.e., acquisition range is maximized when $N_{PIX} = 1$.

The apparent discrepancy in resolution requirements for IR acquisition and recognition can be resolved via detection processing modeled after our understanding of eye/brain processing in human vision [3].

In performing detection of resolved targets in electronic images, humans are apparently capable of performing adaptive integration over the target's area - an astounding feat in light of the fact that the boundaries for area integration must be established **prior** to conscious detection.

The remarkable human capacity for subliminal integration over arbitrarily sized and located target areas is referred to by Rosell as "the perfect synchronous integrator model" of human vision, and is attributed by Rosell to original discovery by Otto Schade, Sr. [4].

The effect of perfect synchronous integration is to enhance by $N_{PIX}^{1/2}$ the single channel SNR, SNR_C, given by Eq. 11. A further characteristic of human vision is that the process of detection cannot be separated from the process of estimating target angular dimensions; the two processes are performed in a coupled and purely parallel fashion.

Figure 4 depicts a conceptual model of the human vision system (HVS) detection process based on a parallel array of spatial matched filters, in which each filter in the array corresponds to a possible target shape. Assuming that the HVS is indeed a perfect synchronous integrator (PSI), every possible target shape is represented in the parallel filter bank, as well as all variants of each shape obtainable by the processes of translation, rotation, and scaling.

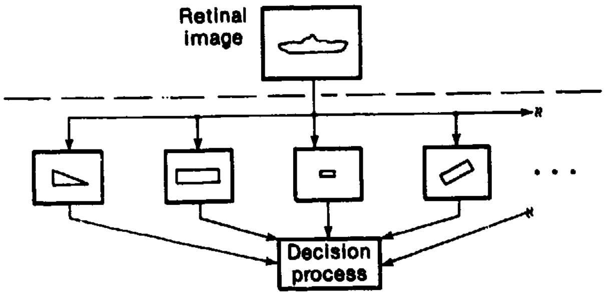


Figure 4. Conceptual model of the human vision system (HVS) detection process, based on an infinite-dimensional array of spatial matched filters.

The detection model diagrammed in Fig. 4 is not directly amenable to simulation or digital realization, since there are an infinite number of filters in the array. We nonetheless fearlessly assume that it is possible to implement with digital hardware an approximation to synchronous area integration, which leads to the definition of a detection SNR,

$$SNR_{det} = \eta \cdot SNR_C \cdot N_{PIX}^{1/2} \cdot \qquad (12)$$

Equivalently, the effect of synchronous area integration can be expressed in terms of a "processing gain"

$$G = (SNR_{det}/SNR_C) = \eta\, N_{PIX}^{1/2} , \qquad (13)$$

presented previously as Eq. 1. Quantity η in Eqs. 12 and 13 is a "processor efficiency" that is included to allow for the possibility of suboptimal processor performance. In the usage of this report, a PSI is any processor, human or machine, characterized by the property $\eta = 1$. By extension, an imperfect synchronous integrator (ISI) is a processor for which $\eta < 1$.

4.0 ACQUISITION RANGE CALCULATIONS

It follows from Eqs. 11 and 12 that

$$\Delta T = \eta^{-1} \cdot SNR_{det} \cdot NE\Delta T / N_{PIX}^{1/2} \ . \qquad (14)$$

Assuming somewhat arbitrarily a threshold value for SNR_{det}

$$SNR_{det}^{T} = 5 \ , \qquad (15)$$

the value of ΔT required for detection, $R\Delta T$

$$R\Delta T = 5 \cdot NE\Delta T / \eta N_{PIX}^{1/2} \qquad (16)$$

is obtained.

Quantity $R\Delta T$ is similar to the Minimum Resolvable Temperature (MRT) parameter conventionally used to describe the performance of human observers of thermal image displays. In fact, $R\Delta T$ can be made equal to MRT if

- SNR_{det}^{T} is set equal to 2.8 (rather than 5), corresponding to the experimentally determined value required by human observers to achieve 50% probability of detection,
- η is set equal to unity, and
- an additional factor $(T_F/0.2)^{1/2}$ is included, where T_F is frame time (typically, 1/30 second), to account for the smoothing of rapid temporal variations by the eye's limited response time (~0.2 s).

The application of MRT-like descriptors to automatic target screeners is also proposed in [21].

In the absence of spatial integration processing (i.e., performing threshold detection directly on the individual channel outputs), Eq. 16 becomes

$$R\Delta T = 5 \cdot NE\Delta T \ . \qquad (17)$$

The ratio of Eqs. 16 and 17 is, of course, Eq. 13.

For fixed sensor IFOV (α and β in Fig. 2), N_{PIX} decreases with increasing range. Hence, from Eq. 16, greater apparent contrast is required to detect distant targets than is required to detect closer targets. Apparent contrast required for simple hot spot detection, given by Eq. 17, is independent of N_{PIX} and thus is also independent of target range.

To further evaluate Eq. 16, the number of pixels on target, N_{PIX}, is expressed as

$$N_{PIX} = A_{ship}/\alpha\beta R^2 \ , \qquad (18)$$

where

A_{ship} = ship's projected area, m^2
α, β = sensor pixel dimensions (cf. Fig. 2), mr
R = range from seeker to ship, km.

From Eqs. 13 and 18

$$G = \eta A_{ship}^{1/2}/R(\alpha\beta)^{1/2} \ . \qquad (19)$$

The apparent contrast required for detection may be written in terms of Eq. 19 as

$$R\Delta T = 5 \cdot NE\Delta T/G \ . \qquad (20)$$

Equations 17, 19, and 20 can be used for calculating seeker acquisition range against ship targets, as indicated graphically in Fig. 5, for the strawman seeker and target characteristics provided in Table 1.

Curve A in Fig. 5 is RΔT for a simple hot spot detection algorithm, from Eq. 17. Curves B and C are RΔT for synchronous integration processing, from Eq. 20, for the front- and beam-aspect targets, respectively.

Also shown in Fig. 5 (curve D) is a representative ship "signature" (i.e., ship apparent contrast

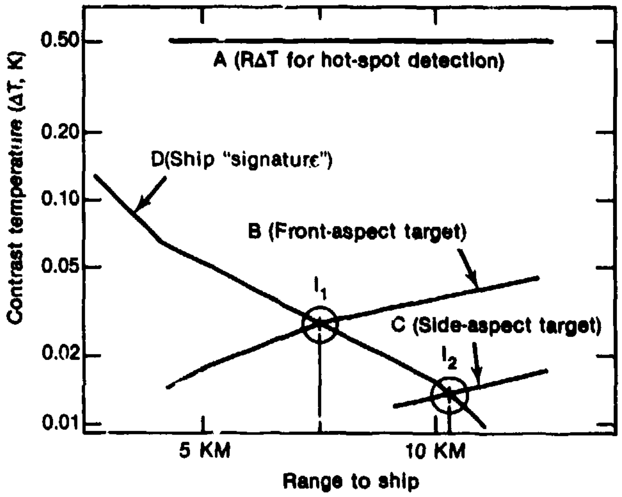


Figure 5. Acquisition range calculations for strawman seeker and target characteristics provided in Table 1.

Table 1
Strawman seeker and target parameters

Seeker Parameter	Meaning	Assumed Value
α	In-scan resolution	0.2 mr
β	Cross-scan resolution	0.2 mr
η	Processor efficiency	1
NEΔT	Temperature sensitivity	0.1 K
Target Parameter	**Meaning**	**Assumed Value**
A_{ship} (90°)	Ship beam-aspect area	4750 m^2
A_{ship} (0°)	Ship bow-aspect area	1100 m^2

temperature vs. range) for a particular locale, season, time of day, probability of weather, sensor height-above-water, and IR waveband [12].

Intersection points I_1 and I_2 in Fig. 5 indicate that synchronous integration processing achieves acquisition ranges of 7.5 and 10.3 kilometers (km) against the front- and beam-aspect targets, respectively. Lack of intersection between curves D and A indicates that hot spot detection fails to achieve target acquisition.

At this juncture, it should be obvious that synchronous area integration offers considerable performance advantage relative to hot spot detection. Less obvious is how these benefits are to be obtained, since the only model presented thus far for a synchronous area integrator (viz., Fig. 4) is unrealizable.

5.0 AN IMPERFECT SYNCHRONOUS INTEGRATOR

In this section we develop the structure of a realizable signal processor (an "imperfect synchronous integrator," or ISI) having performance characteristics almost as good as the unrealizable PSI depicted in Fig. 4.

We start the ISI design with the structure of Fig. 4, i.e., a processor configured as a parallel bank of spatial filters. The PSI is able to accommodate target profile uncertainties in shape, scale, position, and orientation. The first step in reducing processor complexity is to note that the rotational degree of freedom is unneeded in the present application: to sea-skimming missiles flying wings-level, the major axis of ship targets will always appear parallel to the horizon.

The next step in reducing processor complexity is to constrain the spatial filter responses to be rectangles. Thus, when the original image consists of a single bright pixel against an otherwise uniform background, the output image of each filter will be a rectangle of dimensions *k* pixels (cross-scan) by *n* pixels (in-scan). Consequently, each filter in the filter bank can be labeled with its "resolution indices", *k* and *n*, and generally described in terms of "the (k,n) filter." The resulting suboptimal approximation of Fig. 4 is depicted in Fig. 6.

Finally, the required number of filters in the filter bank is further limited by constraining the resolution indices to be powers of 2, between specified minimum and maximum values based on the likely target angular dimensions at initial detection. For example, one may choose

$$\begin{aligned} log_2 k &= 1,\ 2,\ 3 \\ log_2 n &= 2,\ 3,\ \ldots 6\ \ , \end{aligned} \tag{21}$$

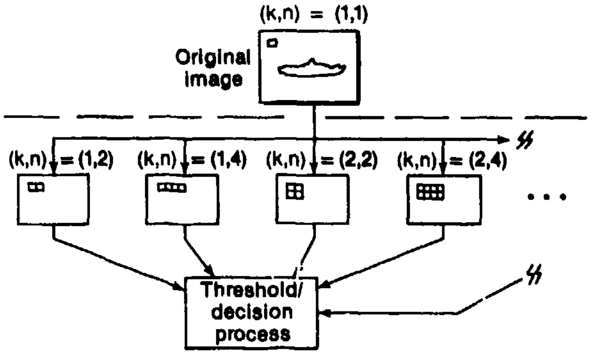


Figure 6. Imperfect synchronous integrator (ISI) obtained from Figure 4 by selecting spatial filter responses as unrotated rectangles.

corresponding to a filter bank comprised of 15 spatial filters. The number of spatial filters may be reduced still further by culling out filters that correspond to unlikely target aspect ratios. As indicated in Fig. 7 for example, there may be very little penalty in removing filters corresponding to $(k,n) = (8,4)$ and $(k,n) = (2,64)$; targets having such aspect ratios can still be detected, although with reduced SNR gain.

The parallel bank of 13 filters corresponding to Fig. 7 may be conceived as having a single image as input and generating 13 blurred/reduced resolution images as its output. (In the image-based simulations discussed in Section 14, a processor having 35 spatial filters was routinely simulated on an IBM PC AT.) It is fortunate from the standpoint of data throughput requirements that the sampling rate for each of the 13 reduced resolution images is substantially lower than for the original full resolution image.

Considering Fig. 6, $n = 4$ is the lowest resolution at which the scene need be observed from the standpoint of matched filtering for detection. This has suggested the structure of a detection preprocessor depicted in Fig. 8. Each of the *I* detector channels is processed by an identical such preprocessor. The notation for image data generally is

$$V_i^{k,n}(j) \tag{22}$$

where we recall that indices (i,j) refer to position in the image (cf. Fig. 1) and indices (k,n) denote the spatial resolution of the image. Thus

$$V_i^{1,1}(j)$$

in Fig. 8 is the image data as originally output from the detector array, at the original $(k,n) = (1,1)$

Azimuth Index, n

Elevation Index, k	4	8	16	32	64
2	2 × 4	2 × 8	2 × 16	2 × 32	
4	4 × 4	4 × 8	4 × 16	4 × 32	4 × 64
8		8 × 8	8 × 16	8 × 32	8 × 64

Figure 7. Matrix of spatial resolutions for a spatial filter array.

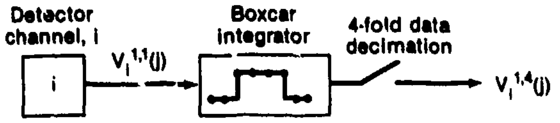


Figure 8. Detection preprocessor block diagram.

resolution. As shown in Fig. 8, the voltage time waveform from detector *i*, denoted

$$V_i^{l,l}(j)$$

(where *j* is time) is smoothed by a boxcar integrator and decimated four-fold in time; i.e., only one of each four samples output from the boxcar integrator is used in subsequent signal processing stages. The effect of the detection preprocessing step is to provide an immediate two-fold improvement in SNR and an immediate four-fold reduction in subsequent data throughput requirements.

Of course, the two-fold SNR improvement is only experienced for targets at least four samples in in-scan extent; smaller targets will have less SNR improvement, with a single-sample target actually experiencing a two-fold degradation in SNR. This drawback is more apparent than real for the following two reasons.

- Subsequent to detection, it is necessary to transfer processing to a classification step. Targets occupying ≤4 samples in-scan are too small to permit image-based classification.
- Size is a key indicator of ship target value. Small ships are generally of less interest as targets than are large ships.

The input/output relationship for the Fig. 8 preprocessor may be expressed as

$$V_i^{1,4}(j) = ¼ \sum_{k=0}^{3} V_i^{1,1}(4j-k) \ , \qquad (23)$$
$$j=1,2,\dots \ [Au/4\alpha] \ .$$

The maximum value of *j* in Eq. 23

$$j_{max} = [Au/4\alpha] \qquad (24)$$

is only one-fourth as large as j_{max} in Eq. 7 due to the four-fold decimation step.
Writing

$$A = \dot{\theta} T_F \qquad (25)$$

where

A = total in-scan field-of-view, mr
$\dot{\theta}$ = angular rate-of-scan mr/s
T_F = frame time, s

and recognizing that the sample rate per channel, f_s (Hz), is given by

$$f_s = j_{max}/T_F \qquad (26)$$

from Eqs. (24)-(26), it can be found that

$$f_s = \dot{\theta} u/4\alpha \qquad (27)$$

The factor 4 in the denominator of Eq. 27 is due to the decimation step in Fig. 8.

The factor (1/4) preceding the sum in Eq. 23 is a normalization that may not be required in a hardware realization of the processor.

The general structure of the complete ISI, which is called a Multi-Resolution Spatial Integrator (MRSI), is shown in Fig. 9. The elements of MRSI are:

- the detection preprocessor (Fig. 8),
- cross-channel averaging (XAV, Section 6),

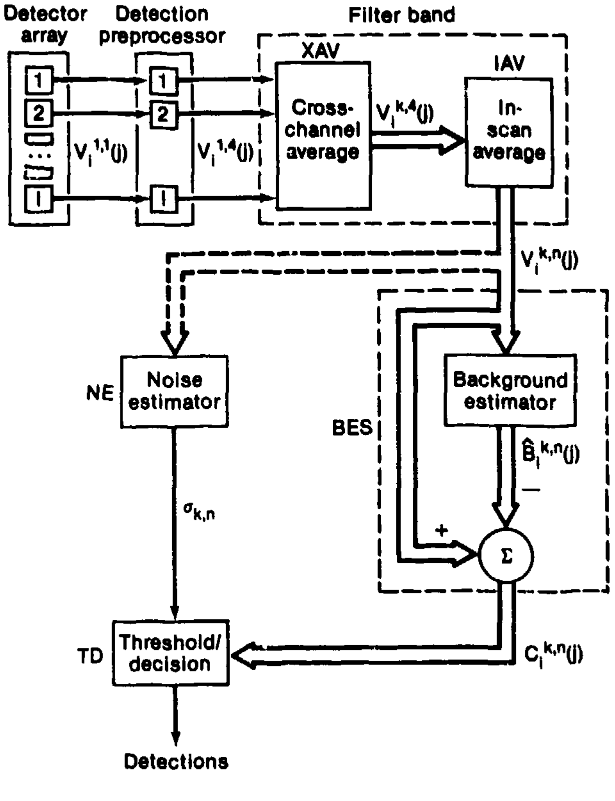


Figure 9. MRSI signal processor block diagram.

- in-scan averaging (IAV, Section 7),
- background estimation and subtraction (BES, Section 9), and
- noise estimation and threshold/decision (NE and T/D, Section 10).

Each capitalized acronym (XAV, IAV, etc.) is the name of the subroutine that implements the corresponding processor function in the FORTRAN simulation (App. A).

The action of the detection preprocessor has already been discussed in connection with Eq. 23 and Fig. 8. The other processing blocks in Fig. 9 will be discussed in subsequent sections of this report.

Although the detector array depicted in Fig. 9 is comprised of square non-staggered elements with a small cross-channel gap, the MRSI processor is readily applied to arrays having all possible combinations of the following attributes: arbitrary values of v; stagger offset between adjacent detector channels; and time-delay-and-integration.

The means for implementing detection preprocessing as in Fig. 8 is an array of I identical processing modules operating simultaneously and uncoupled; i.e., in a purely parallel fashion. A means for implementing an array of spatial filters at varying spatial resolutions is described in Sections 6 through 8. Like the preprocessor, Fig. 8, the multi-resolution processor comprises a large number of simple and identical processing elements operating in parallel. The spatial filter algorithm is separable in in-scan and cross-scan operations, with the cross-scan operations being performed first. Thus, we operate on the detection preprocessor outputs, viz.,

$$V_i^{1,4}(j) \ ,$$

with a "cross-channel averaging processor", XAV, to obtain

$$V_i^{k,4}(j) \ , \ k = 2,4,8$$

which are then provided as input to an "in-scan averaging processor", IAV, to obtain the complete set of reduced resolution images,

$$V_i^{k,n}(j) \ , \quad \begin{aligned} k &= 2,4,8 \\ n &= 4,8,16,32,64 \end{aligned}$$

6.0 CROSS-CHANNEL AVERAGING PROCESSOR (XAV)

The structure of the cross-channel averaging processor, XAV, is shown in Fig. 10. Although three stages of cross-channel processing are required to obtain the three levels of cross-channel resolution indicated in Fig. 7 (k = 2, 4, and 8), for simplicity of illustration, only two of the stages are depicted. Again for simplicity, in Fig. 10, the four quantities A, B, C, and D are defined as follows:

$$\begin{aligned} A &= V_1^{1,4}(j) \\ B &= V_2^{1,4}(j) \\ C &= V_3^{1,4}(j) \\ D &= V_4^{1,4}(j) \end{aligned} \tag{28}$$

Through the first stage of cross-channel averaging, we obtain

$$\begin{aligned} V_1^{2,4}(j) &= (A+B)/2 = ½[V_1^{1,4}(j)+V_2^{1,4}(j)] \quad (29) \\ V_2^{2,4}(j) &= (B+C)/2 = ½[V_2^{1,4}(j)+V_3^{1,4}(j)] \\ V_3^{2,4}(j) &= (C+D)/2 = ½[V_3^{1,4}(j)+V_4^{1,4}(j)] \end{aligned}$$

Generalizing the relationship we see developing in Eq. 29,

$$V_i^{2,4}(j) = ½ \sum_{r=1}^{2} V_{i+r-1}^{1,4}(j) \ , \qquad i=1,2,\ldots\ (I-1) \tag{30}$$

descriptive of the action of the first XAV stage.

Following the XAV second stage logic in Fig. 10 we obtain,

$$V_1^{4,4}(j) = (A+B+C+D)/4 = ¼ \sum_{r=1}^{4} V_r^{1,4}(j) \ , \qquad i=1,2,\ldots\ (I-3)$$

which generalizes to

$$V_i^{4,4}(j) = ¼ \sum_{r=1}^{4} V_{i+r-1}^{1,4}(j) \ , \qquad i=1,2,\ldots\ (I-3) \tag{31}$$

The output of the XAV third stage (not shown in Fig. 10) is found to be

$$V_i^{8,4}(j) = 1/8 \sum_{r=1}^{8} V_{i+r-1}^{1,4}(j) \ , \qquad i=1,2,\ldots\ (I-7) \ . \tag{32}$$

Inspection of Eqs 30 through 32 indicates the general input/output description of XAV can be written as

$$V_i^{k,4}(j) = 1/k \sum_{r=1}^{k} V_{i+r-1}^{1,4}(j) \ , \qquad k=2,4,8 \qquad i=1,2,\ldots\ (I-k+1) \tag{33a}$$

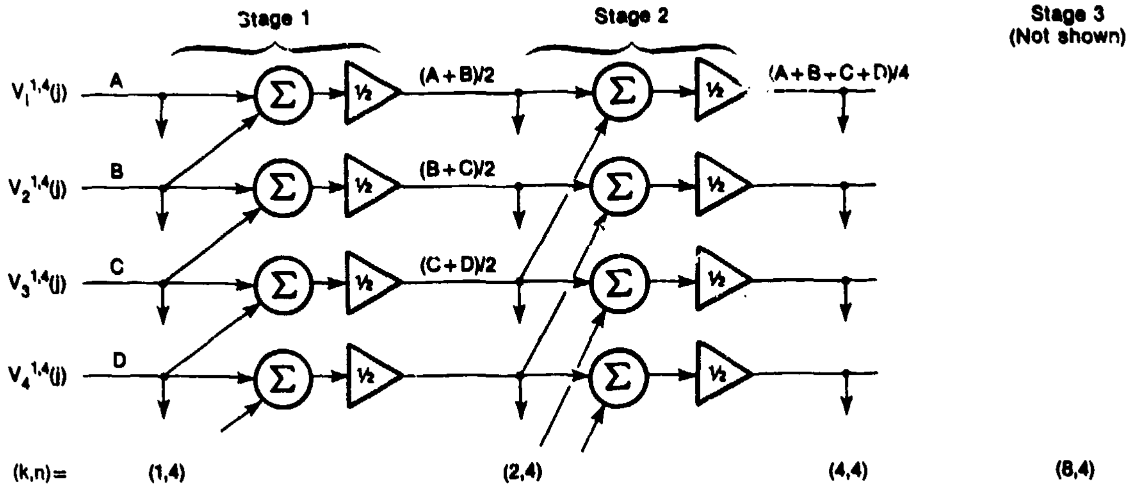


Figure 10. Cross-channel averaging processor, XAV. Inputs to XAV, denoted $V_i^{1,4}(j)$, are provided by detection preprocessor (Figure 8). Outputs of XAV, denoted by unterminated arrows pointing downward, are provided to IAV for further processing.

Cross-channel averaging successively reduces the number of elevation channels (and hence the maximum value of i) at progressively higher resolutions

$$I(k) \equiv max(i) = I\text{-}k+1 \ , \qquad (34)$$

where the number of elevation channels at the original resolution, I, is given by Eq. 6.

It follows from Eq. 33 that cross-channel processing is instantaneous, i.e., determination of $V^{k,4}$ at time j depends only on values of $V_m^{1,4}$, also at sampling time j.

The position of XAV within the overall processing structure is indicated in Fig. 9.

Equivalent to Eq. 33a, we can write

$$V_i^{2k,4}(j) = \tfrac{1}{2}[V_i^{k,4}(j) + V_{i+k}^{k,4}(j)] \ , \qquad (33b)$$
$$k=1,2,4$$
$$i=1,2,\ldots\ (I-2k+1) \ .$$

The FORTRAN routine XAV (App. A) implements Eq. 33b.

7.0 IN-SCAN AVERAGING PROCESSOR (IAV)

The in-scan averaging processor, (IAV), is comprised of a large number of identical processing "sections", each structured as shown in Fig. 11. Since a separate IAV section is required for each distinct pair of values (i,k), the total number of required sections (each identical to Fig. 11) may be computed as

$$N_{IAV} = \sum_{k=2,4,8} I(k)$$

With $I(k)$ given by Eq. 34,

$$N_{IAV} = \sum_{k=2,4,8} (I-k+1) = 3I-11 \ . \qquad (35)$$

For example, if the scanning seeker has

$$I = 128 \qquad (36)$$

detector channels, from Eq. 35, it is found that

$$N_{IAV} = 373 \qquad (37)$$

IAV sections, each identical to Fig. 11, will be required.

More generally, the number of required IAV sections may be simply estimated as I times the number of required cross-scan averaging stages.

The action of IAV is developed with reference to Fig. 11 as

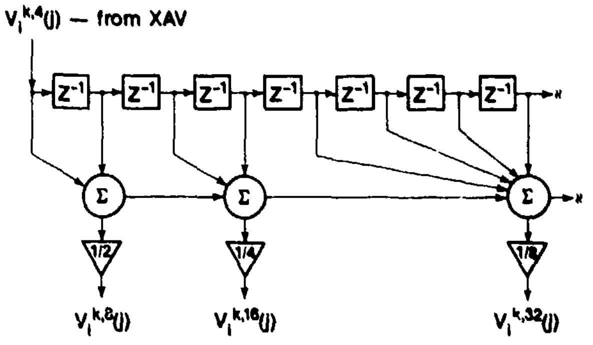


Figure 11. Transversal filter realization of one section of the in-scan averaging processor, IAV. Inputs to IAV are provided by the cross-scan averager (XAV). A large number of such sections, operating simultaneously and uncoupled, comprise the complete IAV. Output of IAV is an array of reduced resolution images. Processing elements denoted as z^{-1} are unit delays. Final 8 delay elements and final summing node are not shown.

$$V_i^{k,8}(j) = \tfrac{1}{2} \sum_{s=1}^{2} V_i^{k,4}(j-s+1) \qquad (38)$$
$$V_i^{k,16}(j) = \tfrac{1}{4} \sum_{s=1}^{4} V_i^{k,4}(j-s+1)$$
$$V_i^{k,32}(j) = 1/8 \sum_{s=1}^{8} V_i^{k,4}(j-s+1) \ ,$$

which generalizes readily as

$$V_i^{k,4n_o}(j) = 1/n_o \sum_{s=1}^{n_o} V_i^{k,4}(j-s+1) \ , \qquad (39)$$
$$n_o = 2,4,8,16$$
$$j = n_o,\ (n_o+1),\ (n_o+2),\ldots \ .$$

If the maximum and minimum values of in-scan resolution index are denoted, respectively, as n_{max} and n_{min}, the number of delay elements in each IAV section (Fig. 11) is given by

$$N_{delays} = (n_{max}/n_{min}) - 1 \ . \qquad (40)$$

For the example of Eq. 21

$$N_{delays} = (64/4) - 1 = 15 \ , \qquad (41)$$

delay elements per IAV section are obtained. For this example, IAV provides outputs corresponding to the following five in-scan resolutions

$$n = 4,8,16,32,64 \ . \qquad (42)$$

The number of required delay elements can become quite large for IAV sections structured as per Fig. 11. For example, if instead of Eq. 42

$$n = 1,2,4,8,16,32,64,128, \qquad (43)$$

the number of delay elements per IAV section increases from 15 to 127 in spite of the fact that the number of in-scan resolutions has only been increased from 5 to 8.

Depicted in Fig. 12 is an alternative structure for the IAV sections, modified from Fig. 11 to reduce the required number of delay elements. The processing element labeled "A" in Fig. 12 is structured like the original IAV section, Fig. 11. Each of the boxes labeled "B" in Fig. 12 is an infinite-impulse-response (IIR) approximation to Eq. 39 for the appropriate value of n_o.

Particularly simple IIR structures, shown in Figs.

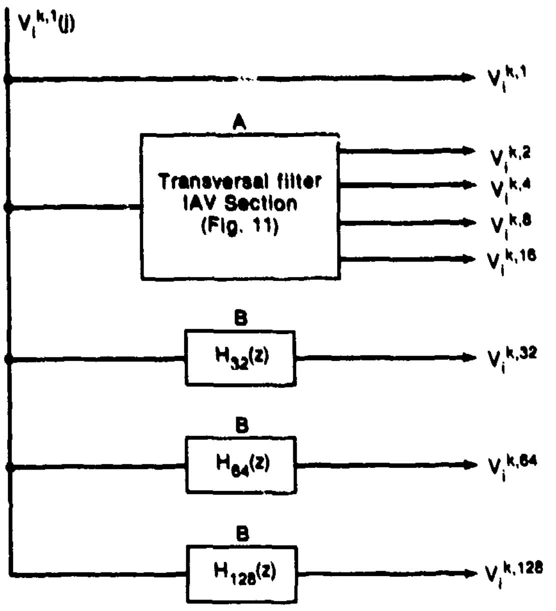


Figure 12. IAV section, modified to minimize the number of required delay elements.

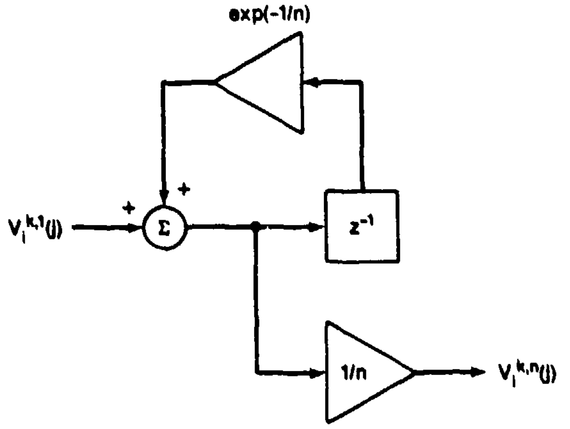


Figure 13. Suboptimal approximation of $H_n(z)$ requiring just one memory/delay element.

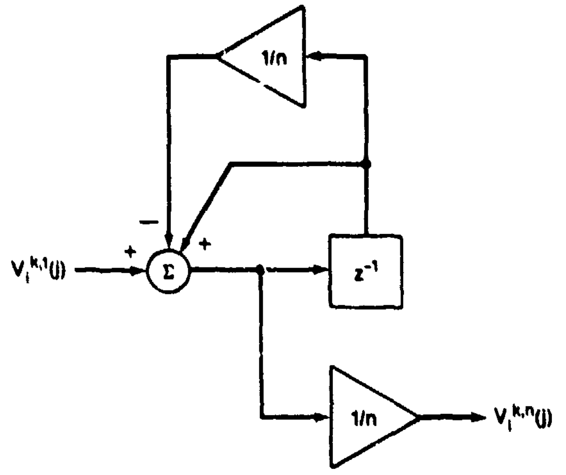


Figure 14. Approximate realization of $H_n(z)$, Figure 13. Since n is a power of 2, both fixed gain amplifiers are realizable as bit shifts.

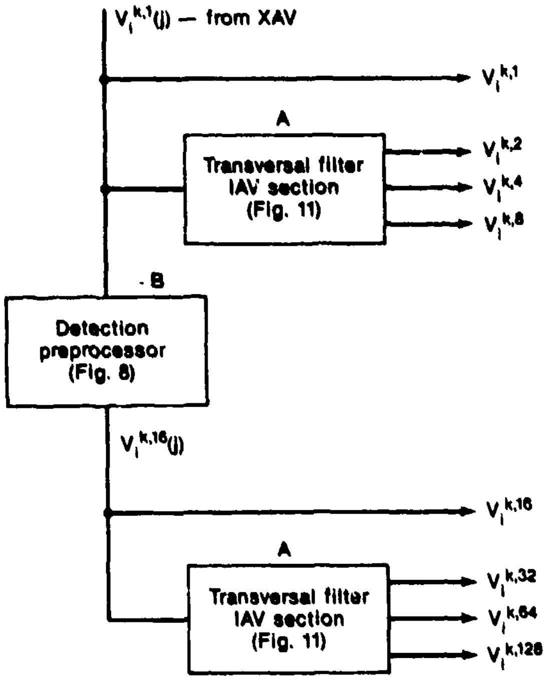

$V_i^{k,1}(j)$ — from XAV
$V_i^{k,1}$
A
Transversal filter IAV section (Fig. 11)
$V_i^{k,2}$
$V_i^{k,4}$
$V_i^{k,8}$
B
Detection preprocessor (Fig. 8)
$V_i^{k,16}(j)$
$V_i^{k,16}$
A
Transversal filter IAV section (Fig. 11)
$V_i^{k,32}$
$V_i^{k,64}$
$V_i^{k,128}$

Figure 15. Alternative IAV structure.

13 and 14, require just one sample delay per filter function, $H_n(z)$. Following this approach, the IAV section corresponding to the eight in-scan resolutions given by Eq. 43 can be implemented with just 18 delay elements rather than 127 elements, as would be required by the original approach (Fig. 11) alone.

Of course, the reduced number of processing elements achieved via IIR processing (Fig. 13) relative to finite-impluse response (FIR) processing (Fig. 11) is at the expense of reduced SNR. A relevant discussion of this point is provided in [22]. The SNR penalty involved in developing IIR approximations to Eq. 39 depends on the order of the IIR filter; procedures for designing IIR structures subject to design criteria appropriate to the present application are given in [23, Chap. 8] and [24].

Yet another approach to designing IAV sections to minimize the number of processing elements is shown in Fig. 15. The components labeled "A" in Fig. 15 are structured as per Fig. 11; the component labeled "B" in Fig. 15 is structured like the detection preprocessor (Fig. 8/Eq. 23), but modified for 16-fold resolution reduction and decimation-in-time.

The development of a multi-resolution spatial filter array via the structure of Figs. 10 through 15 is an original contribution of the work described in this report. Some virtues of this approach are that

- The image data are processed sequentially by column; since scanned linear arrays acquire imagery sequentially by column, the processing approach described eliminates the need to buffer large amounts of data; and
- The total processor comprises a large number of simple, identical, modules, operating in parallel, and thus appears inherently well-suited to real-time applications.

8.0 THE PARALLEL BANK OF SPATIAL FILTERS

The multi-resolution filter bank is realized by concatenating XAV and IAV, as shown in Fig. 9. A single input/output relationship for these two combined filters is obtained by substituting Eq. 33 into Eq. 39, with the result

$$V_i^{k,4n_o}(j) = (1/kn_o) \sum_{s=1}^{n_o} \sum_{r=1}^{k} V_{i+r-1}^{1,4}(j-s+1) \text{ ,} \quad (44)$$

which may be written in a compressed notation as

$$V^{k,4n_o} = V^{1,4} \circledast rect(k,n_o) \text{ ,} \quad (45)$$

where

$V^{1,4} = \{ V_i^{1,4}(j) \}_{ij}$ = image at output of detection preprocessoi, Fig. 8

$V^{k,4n_o} = \{ V_i^{k,4n_o}(j) \}_{ij}$ = reduced resolution image with indices $(k, 4n_o)$

$rect(k, n_o)$ = 2-D rectangle function, of dimensions $k \times n_o$ and amplitude $(1/kn_o)$.

The meaning of Eqs. 44 and 45 is explained with the aid of Fig. 16. As shown in the figure, the effect of image convolution with a rectangular smoothing

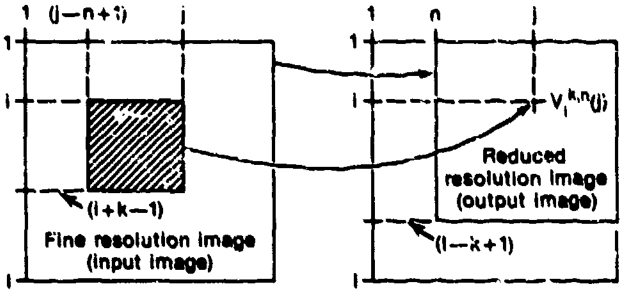
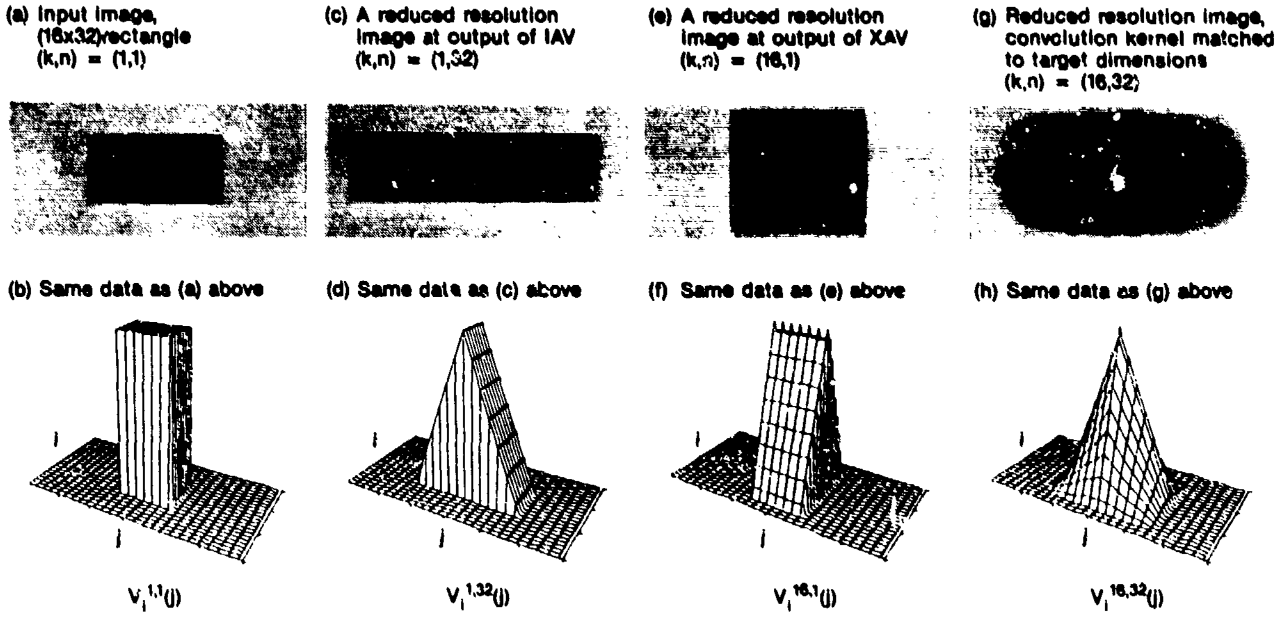


Figure 16. Convolution with a rectangular kernel may be interpreted as a mapping from a fine resolution image (Figure left) to a reduced resolution image (Figure right). The (i,j) — pixel value in the reduced resolution image is obtained as the average of the pixel values within a rectangular region (shaded box) of dimensions k x n, whose upper-right-corner is located at the (i,j) — pixel location in the input image. As shown in the Figure, resolution reduction produces some image shrinking.


(a) Input image, (16x32)rectangle (k,n) = (1,1)

(c) A reduced resolution image at output of IAV (k,n) = (1,32)

(e) A reduced resolution image at output of XAV (k,n) = (16,1)

(g) Reduced resolution image, convolution kernel matched to target dimensions (k,n) = (16,32)

(b) Same data as (a) above

(d) Same data as (c) above

(f) Same data as (e) above

(h) Same data as (g) above

Figure 17. Rectangle test image, at original resolution, (a) — (b), and at 3 reduced resolutions, (c) — (h). Data are prior to background subtraction.

function is to perform a mapping from a fine resolution image to a reduced resolution/smoothed image, in which the (i,j)-pixel value in the reduced resolution image is obtained as the average of the pixel values within a rectangular region of dimensions $k \times n_o$, whose upper-right-hand corner is located at the (i,j)-pixel location in the input image.

The XAV/IAV spatial filter array was implemented as a FORTRAN program and exercised against several simple test targets to confirm correct operation. Figure 17 depicts the results of one such simulation, showing the appearance of a 16×32 rectanguiar test image at three reduced resolutions. Detailed numerical evaluations of similar output images for bar and point targets (for which the convolution products in Eq. 45 are exactly calculable) were used to verify correct functioning of the FORTRAN simulation.

To illustrate the SNR-enhancing effect of spatial filtering, white gaussian noise was added to the rectangle test image to reduce the input image SNR (target contrast/RMS noise) to unity, i.e.,

$$SNR_C = 1 \ .$$

The result is shown as the upper-left image in Fig. 18. Numerical analysis of the images output from IAV indicates that SNR is enhanced predictably as a function of spatial resolution. In particular, the (16,32)-filter output, shown bottom-left in Fig. 18, is found to have SNR very nearly equal to $(16\times32)^{1/2} = 16\sqrt{2}$. Also shown for illustrative purposes in Fig. 18 is the effect of applying spatial filtering to a frigate silhouette.

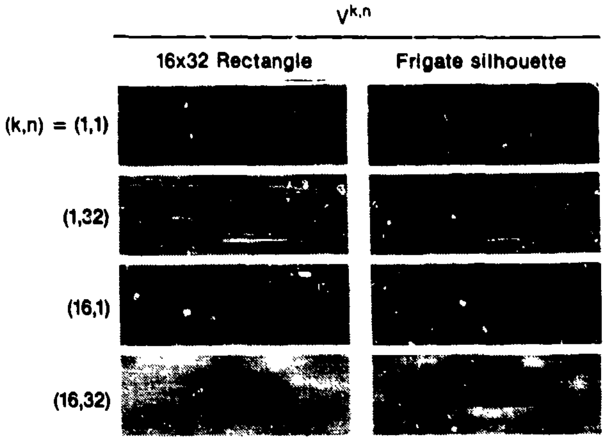


Figure 18. Multiresolution processing enhances SNR. Four images (left) are for rectangle input image; four images (right) are for frigate silhouette. Images at original resolution, having SNR = 1, are in the top row (rectangle is top-left; frigate is top-right).

9.0 BACKGROUND ESTIMATION AND SUBTRACTION (BES)

The need for background estimation and subtraction (BES) is explained with the aid of Fig. 19. The detection problem is conceptualized as being one-dimensional in Fig. 19 to simplify the discussion.

Figure 19 depicts a target of size γ_s pixels observed against three types of backgrounds, viz., a constant background of unknown intensity (part a); a background of constant but unknown intensity gradient (part b); and a "slowly varying" background (part c). Successful detection requires that the processor establish an intensity threshold, $T(j)$, that rides above the unknown and possibly varying background, $B(j)$, by a fixed intensity offset, Ω.

Figure 20 depicts a conceptual means for establishing the threshold and accomplishing detection as per the examples in Fig. 19. Estimating under the null target hypothesis what the background would have been at each position j, $\hat{B}(j)$, appears to be an essential element in developing a threshold $T(j)$ for target detection against spatially varying backgrounds.

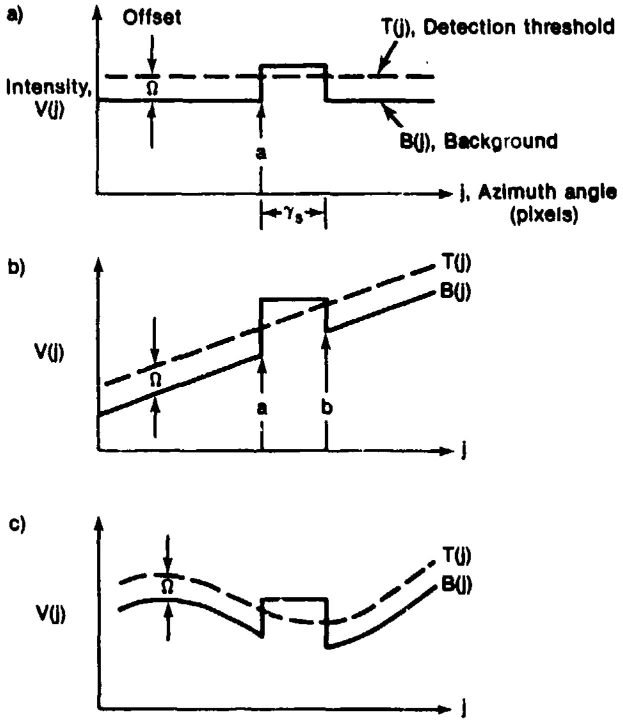


Figure 19. Examples of 1-D detection problems illustrating the need for BES. Part a), target of dimension γ_S (pixels) in constant background. Level 'a' is unknown. Part b), target embedded in constant slope background. Background parameters 'a' and 'b' are unknown. Part c), target observed against slowly varying background.

Equivalent to Fig. 20, the detection process can be conceived as being accomplished in two stages, viz., background estimation and subtraction (BES)

$$C(j) = V(j) - B(j) \ , \qquad (46)$$

followed by threshold detection

$$C(j) \gtrless \Omega \qquad (47)$$

A process analogous to Eq. 46 is separately applied to all waveforms, i.e.,

$$C_i^{k,n}(j) = V_i^{k,n}(j) - \hat{B}_i^{k,n}(j) \ , \qquad (48)$$

as shown in Fig. 9.

A number of candidate filters for developing the background estimate, $\hat{B}$, is discussed next.

Assuming that the direction of scan is parallel to the horizon, the background estimate may be developed as

$$\hat{B}_i^{k,n}(j) = V_i^{k,n}(j-nu) \ . \qquad (49)$$

In the event that the background level is found to contain appreciable variation over in-scan angles $\approx n\alpha$, an alternative background estimator is

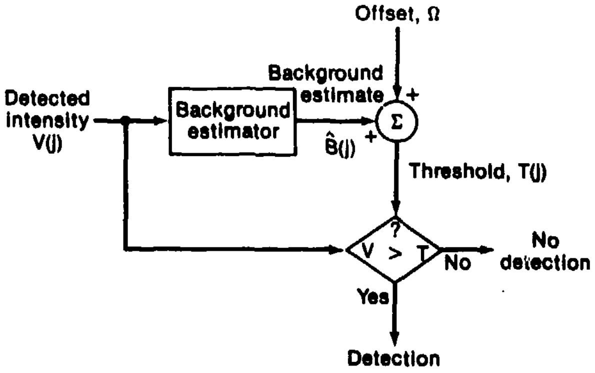


Figure 20. Conceptual signal processor for implementing threshold detection as per Figure 19 examples.

$$\hat{B}_i^{k,n}(j) = [V_i^{k,n/2}(j-5nu/4) + V_i^{k,n/2}(j+5nu/4)]/2. \quad (50)$$

The image areas ("windows") used by the background estimators, Eqs. 49 and 50, are depicted in Fig. 21. The "window gaps" in Fig. 21b, separating the background windows from the target window, are intended to prevent target self-thresholding (with a consequently degraded SNR) when the target's in-scan extent, γ_s, over-spills the target window; i.e., when

$$n\alpha < \gamma_s \leq 2n\alpha . \quad (51)$$

The variance of a random process $X(t)$ is denoted as

$$Var\ (X) = E\{[X-E(X)]^2\} , \quad (52)$$

where $E\{\cdot\}$ indicates expectation over sensor noise statistics.

For spatially uniform backgrounds, specifically excluding the possibility of a target in the scene, we define

$$\sigma^2_{kn} = Var[V_i^{kn}(j)] , \quad (53)$$

where the presumed lack of dependence of $\sigma^2_{k,n}$ on i is discussed below.

Since $V_i^{k,n}(j)$ is statistically independent of $\hat{B}_i^{k,n}(j)$ for all background estimators of interest, it follows from Eqs. 48 and 53 that

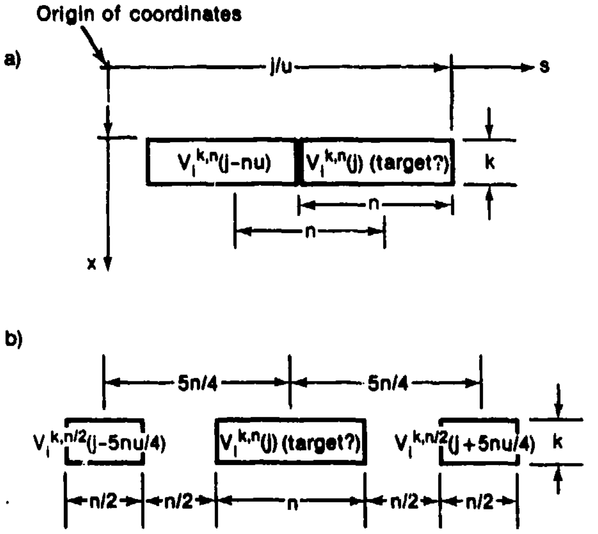


Figure 21. Image areas used by two background estimators. Part a) corresponds to Eq. (49); part b) corresponds to Eq. (50). Cross-channel (x) dimensions are in channel units; in-scan (s) dimensions are in dwells. Scan direction (s) is parallel to the horizon.

$$Var[C_i^{kn}(j)] = \sigma^2_{kn} + Var[\hat{B}_i^{kn}(j)] . \quad (54)$$

It follows from Eqs. 49 and 50 that, for both of these background estimators,

$$Var[\hat{B}_i^{kn}(j)] = \sigma^2_{kn} . \quad (55)$$

Thus, from Eqs. 54 and 55,

$$Var[C_i^{kn}(j)] = 2\sigma^2_{kn} , \quad (56)$$

i.e., for both background estimators considered thus far, the differencing process used to develop estimated contrast degrades two-fold the variance of the waveforms.

A relatively less noisy contrast can be developed by using a larger background window; e.g.,

$$\hat{B}_i^{kn}(j) = V_i^{k,2n}(j-3nu/2) , \quad (57)$$

depicted in Fig. 22. It follows from Eq. 57 that

$$Var[\hat{B}_i^{kn}(j)] = \sigma^2_{kn}/2 . \quad (58)$$

and from Eqs. 58 and 54 that

$$Var[C_i^{kn}(j)] = 1.5\sigma^2_{kn} . \quad (59)$$

Comparing Eqs. 56 and 59, we see that the background estimate provided by Eq. 57 results in a lower-noise contrast estimate.

The use of larger background windows (e.g., Fig. 22) provides improved acquisition performance against unstructured backgrounds at the expense of reduced performance against structured ("cluttered") backgrounds. The trade-offs involved in selecting the MRSI background estimator window dimensions are in many respects analogous to those encountered in designing signal processors for detecting spatially unresolved targets, commonly known as infrared search and track (IRST) devices [25].

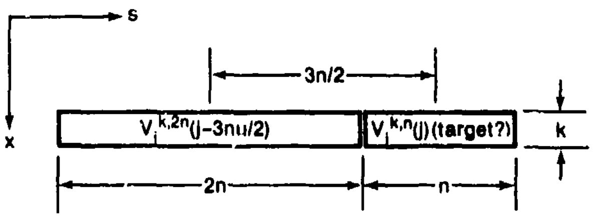


Figure 22. Image area used by low-noise background estimator, Eq. (57). Scan direction (s) is parallel to the horizon.

Developing the background estimate for parallel-to-horizon scanning via Eq. 49, 50, or 57 imposes an additional memory requirement beyond that already required by IAV.

Assuming that the direction of scan is perpendicular to the horizon, the background estimator analogous to Eq. 49 is

$$\hat{B}_i^{kn}(j) = V_{i+k}^{kn}(j) \ . \qquad (60)$$

Image areas relevant to Eq. 60 are depicted in Fig. 23.

Analogous to Eqs. 50 and 57, one now has for cross-horizon scanning

$$\hat{B}_i^{kn}(j) = [V_{i-k}^{k/2,n}(j) + V_{i+3k/2}^{k/2,n}(j)]/2 \ , \qquad (61)$$

and

$$\hat{B}_i^{kn}(j) = V_{i+k}^{2k,n}(j) \ . \qquad (62)$$

Developing background estimates for cross-horizon scanning via Eq. 60, 61, or 62 imposes no additional memory requirement beyond that already required by IAV.

It should be noted that the need for implementing background suppression processing, and the likely accuracy of any particular background estimator, is entirely dependent on the spatial structure of the ocean's thermal emission. The sea radiance is generally a function of many variables, including wind speed, sea height, nadir angle of the viewing line-of-sight, viewer's height above water, cloud cover, sunglints, and surface slicks [26].

In the processor simulation (App. A), the background estimator described by Eq. 50 and Fig. 21b, performing linear trend removal from the background, was chosen. Linear (or nearly-linear) trends may be caused by instrumental effects as well as by true thermal gradients in the scene.

Figure 24 provides an example of a measured thermal image in which the sea background appears to display a nearly linear trend with azimuth. It cannot be said with certainty whether the trend observed in this data is due to an instrumental effect or scene thermal variation. Also, the magnitude of the apparent change in background level from one end of the scan to the other is, in this case, still much smaller than the ship's contrast (cf. Fig. 32, top).

Continuing the example that led to Fig. 17, Fig. 25 depicts a rectangular test target as seen at four resolutions, after background subtraction. The images in Fig. 25, like those in Fig. 17, are noise free.

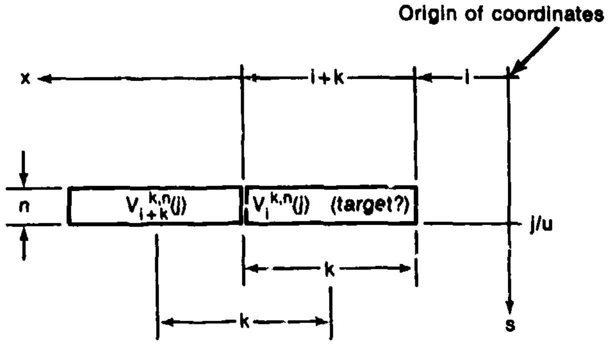


Figure 23. Image area used by background estimator, Eq. (60), for scan direction (s) perpendicular to the horizon.

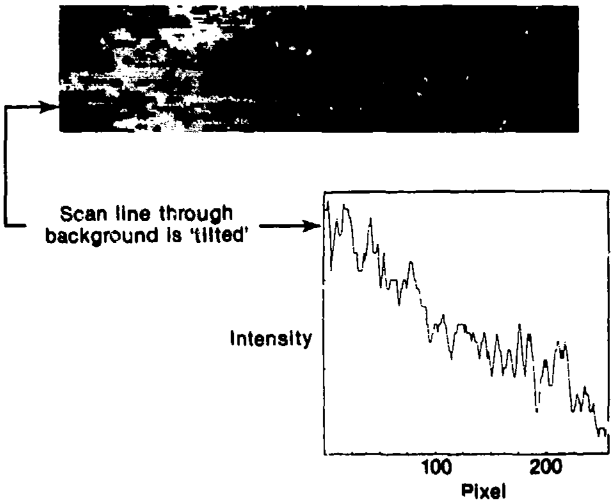


Figure 24. Measured thermal image of a ship observed against a sea background, in which a scan line through the background shows a nearly linear trend.

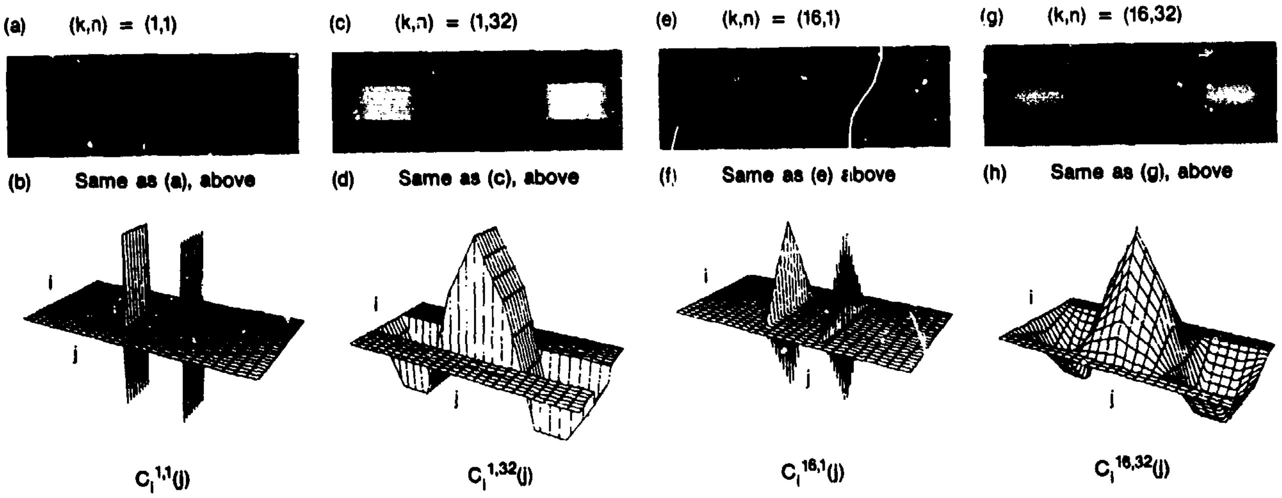


Figure 25. Continuation of example from Figure 17, depicting rectangular test target at 4 resolutions, after background subtraction.

10.0 THRESHOLD/DECISION (TD) AND NOISE ESTIMATION (NE)

Threshold/decision processing (TD) currently implemented in the simulation software (App. A) is based on performing a forced-choice detection in each frame. Thus, rather than perform the threshold comparison indicated in Eq. 47, our TD processor takes the form shown in Fig. 26. Each contrast image

$$\{C_i^{k,n}(j)\}_{ij}$$

is normalized with respect to a noise standard deviation, $\sigma_{k,n}$ to obtain a "signal-to-noise image". The entire set of normalized contrast images is then examined for a global maximum to provide the forced-choice detection. No image storage is required to implement TD, since the search for the "brightest pixel" is performed one column at a time. As a practical matter

$$\max_{i,j,k,n} [C_i^{k,n}(j)] = \max_j W(j) \ , \qquad (63)$$

is calculated, where

$$W(j) = \max_{i,k,n} [C_i^{k,n}(j)] \ . \qquad (64)$$

The "high data rate" part of the search process is contained in Eq. 64. For the set of 13 spatial resolutions given by the example of Fig. 6, and assuming that the seeker has I=128 detector channels (as per Eq. 36), it can be shown that Eq. 64 implies that, at every fourth sampling instant, TD must find the largest of N_W values, where

$$N_W = 4I(2) + 5I(4) + 4I(8) \ ,$$

with $I(k)$ given by Eq. 34. Thus, implementation of Eq. 64 requires that on every fourth sampling instant, TD find the maximum of 1617 data values.

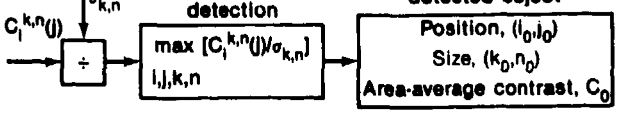


Figure 26. Threshold/decision (TD) logic for forced-choice detection.

TD can be modified simply to accommodate multiple target detections within a large field-of-view by implementing Eq. 63 as a search for local maxima in $W(j)$ - so long as the various targets are spatially well-separated in azimuth.

The action of the noise estimation (NE) component of Figs. 9 and 26 is discussed next. The purpose of NE is to establish estimates for the noise variances $\sigma^2_{k,n}$ defined in Eq. 53.

The root-mean-square (RMS) noise levels of the individual cross-scan channels are denoted as

$$\sigma_i, \quad i = 1,2,\ldots, I \ ,$$

where σ_1 is the RMS value of an additive noise component of waveform $V_1(j)$, σ_2 corresponds similarly to $V_2(j)$, etc. Assuming that the noise level is about the same for each channel

$$\sigma_i \approx \sigma_o \ , \ i=1,2,\ldots,I \ , \qquad (65)$$

and that the additive noise is signal-independent,

$$\partial\sigma_i/\partial V_i = 0 \ , \qquad (66)$$

the RMS noise of waveform $V_i^{kn}(j)$ at the output of IAV is given by

$$\sigma_{kn} = \sigma_o/(kn)^{1/2} \ . \qquad (67)$$

If the different cross-scan channels have appreciably different noise levels, invalidating Eq. 65, it may then be desirable to include a noise calibration mode in the signal processor. This entails illuminating the focal plane uniformly (e.g., by staring into the radiant source used also for dc-restoration), and then using the sample variance of $V_i^{kn}(j)$ as an estimate for $\sigma^2_{kn}(i)$. The RMS noise, σ_{kn}, in Eq. 67 is then replaced by $\sigma_{kn}(i)$.

The possibility of implementing a noise calibration mode (as an alternative to implementing a "hardwired" calibration, Eq. 67), is indicated in Fig. 9 as a dashed connection between IAV and NE. Our software simulation (App. A) does not in fact use Eq. 67. At some expense in execution time, the simulation prior to each forced-choice detection executes a calibration run against a noise-only frame, calculating values of $\sigma_{k,n}$ as sample variances, rather than by using the closed form expression, Eq. 67, that is valid only for white noise.

11.0 PROCESSOR SIGNAL-TO-NOISE MODEL

In this section, we complete the simple processor signal-to-noise model (cf. Eqs. 1, 13, and 19)

$$G = \begin{cases} \eta N^{1/2}_{PIX} \\ \eta A^{1/2}_{ship}/R(\alpha\beta)^{1/2} \end{cases} , \qquad (70)$$

by showing that the choice in Section 5.0 of rectangular response functions provides a simple geometrical basis on which to calculate the "processor efficiency", η.

The geometry of the calculation is depicted in Fig. 27, in terms of two locally defined quantities, A and B, defined as follows:

A = number of target pixels within a rectangular region of integration

B = number of background/non-target pixels within the rectangular region of integration.

The contributions of quantities A and B relevant to calculating PSI and ISI SNR performance are given in Table 2. For example, the increase in RMS noise is always equal to the square root of the number of pixels within the area of integration, which for PSI (i.e., perfect) processing is simply the ship's area (defined in Eq. 70 as N_{PIX}), and which for ISI (i.e., imperfect) processing is the area of the rectangular response function,

$$kn = (A+B) .$$

The third column in Table 2, SNR gain (G), is obtained simply as the ratio of the entries in the first and second columns.

The processor efficiency, η, is calculated by ratioing the third column entries from Table 2,

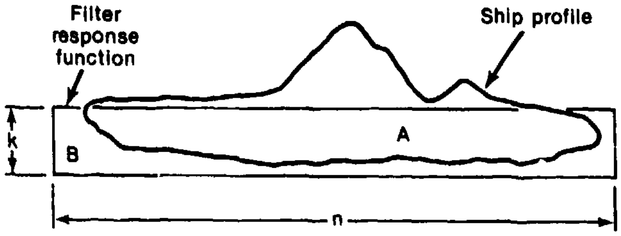


Figure 27. Geometry for calculating processor efficiency, η, entering into Eq. (70).

Table 2
Contribution of quantities A and B, from Fig. 27, relevant to calculating PSI and ISI SNR performance

Processor	Signal Enhancement	RMS Noise Increase	SNR Gain (G)
Perfect Synchronous Integrator (PSI)	N_{PIX}	$N^{1/2}_{PIX}$	$N^{1/2}_{PIX}$
Imperfect Synchronous Integrator (ISI)	A	$(kn)^{1/2}$	$A/(kn)^{1/2}$

$$\eta = G(ISI)/G(PSI) = A/(kn \cdot N_{PIX})^{1/2} . \qquad (71)$$

Equation 71 is the desired expression for processor efficiency, η, which, taken together with Eq. 70, comprises what we call the processor SNR model.

We illustrate the application of Table 2 and Eq. 71 with an example, in which it is assumed that the target dimensions are one pixel in elevation and m pixels in azimuth and that each of the m target pixels has the same brightness. Also, we uniquely define an integer, n_o, in terms of m, as

$$n_o \leq m < 2n_o , \qquad (72)$$

where n_o is an integer power of 2. It follows from Eq. 72 that the MRSI processor will estimate the target's length as being either n_o or $2n_o$ pixels, depending on which of the $(k,n) = (1,n_o)$ or $(k,n) = (1,2n_o)$ filter output amplitudes is largest. The entries in Table 3 are calculated from the second row of Table 2 (ISI), by setting $k=1$ and $A=n_o$ (when $n=n_o$) or $A=m$ (when $n=2n_o$). The processor gain, G, is obtained from Table 3 as

$$G(\hat{m}_1) = \max_n G(n) = \max_n [n_o^{1/2}, m/(2n_o)^{1/2}], \qquad (73)$$

where $\hat{m}_1$ is an initial processor-derived estimate for target size.

It follows also from Table 2 (first row, last column) that

$$G(PSI) = m^{1/2} . \qquad (74)$$

Taking the ratio of Eqs. 73 and 74, we obtain the processor efficiency

Table 3
Quantities relevant to calculating MRSI processor SNR gain (G) and efficiency (η) for the example discussed in connection with Eq. 72. Quantity A is defined as in Fig. 27.

n	A	$G(n)=A/n^{1/2}$	$\eta(n)=A/(nm)^{1/2}$
n_0	n_0	$n_0^{1/2}$	$(n_0/m)^{1/2}$
$2n_0$	m	$m/(2n_0)^{1/2}$	$(m/2n_0)^{1/2}$

$$\eta = \eta(\hat{m}_1) = G(\hat{m}_1)/m^{1/2} \qquad (75)$$

where the quantities $\eta(n)$ are given in Table 3.

This example, Eqs. 72 through 75, is developed numerically in Table 4 for target dimensions

$$1 \leq m \leq 32,$$

some results of which are plotted in Figs. 28 and 29.

The SNR gain, Fig. 28, has a step-stair variation with target size, intervals of constant gain alternating with intervals of constant slope. The breakpoints in Fig. 28 may be calculated by setting equal the quantities $G(n_o)$ and $G(2n_o)$ from Table 3

$$n_o^{1/2} = m/(2n_o)^{1/2} \Big|_{m=m_r}, \quad n_o = 1,2,4,8,16,\ldots \quad .$$

Thus, the Fig. 28 breakpoints occur at non-integer values of m, denoted by m_r, and given by

$$m_r = 2^{r-1/2}, \quad r = 1,2,3,4,\ldots \quad . \qquad (76)$$

Using vertical bar targets in Gaussian noise, we have validated Eq. 73 using the MRSI simulation provided in App. A. In performing this simulation, we used an input SNR of unity, $SNR_C = 1$, and an elevation field-of-view of I=64 pixels. Estimating the SNR gain, G, as the average value of 64 values obtained for the array of 64 elevation channels, the results appear exactly as shown in Fig. 28; i.e., the gain values obtained from Eq. 73 and those obtained via simulation are indistinguishable on the scale of Fig. 28.

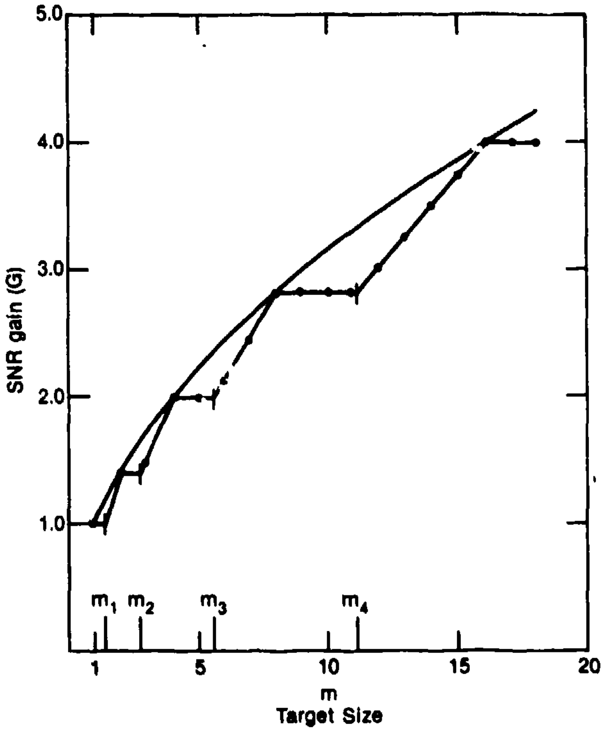


Figure 28. SNR gain from Table 4. Solid curve, G(PSI) = $m^{1/2}$; dotted values, G(ISI) from Table 4. Values of breakpoints m_r given by Eq. 76.

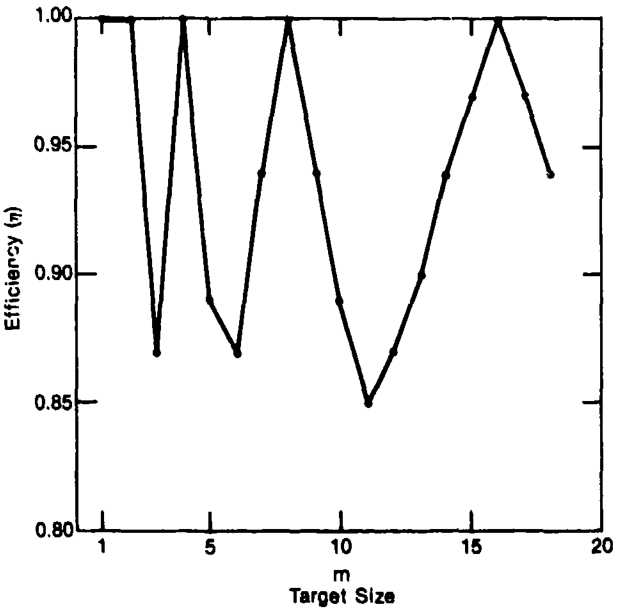


Figure 29. SNR efficiency, from Table 4. Theoretically perfect performance corresponds to $\eta = 1$.

Table 4

Calculation of MRSI processor gain (G) and efficiency (η) as a function of target size (m) for the example discussed in connection with Eqs. 72 through 75

m	n_0	$2n_0$	$\max\{ n_0^{1/2}$,	$m/(2n_0)^{1/2}\}$	$= G(ISI)$	$\eta = G/m^{1/2}$	$\hat{m}_1$
1	1	2	1.00	0.71	1.00	1.00	1
2	2	4	1.41	1.00	1.41	1.00	2
3	2	4	1.41	1.50	1.50	0.87	4
4	4	8	2.00	1.41	2.00	1.00	4
5	4	8	2.00	1.77	2.00	0.88	4
6	4	8	2.00	2.12	2.12	0.87	8
7	4	8	2.00	2.47	2.47	0.94	8
8	8	16	2.83	2.00	2.83	1.00	8
9	8	16	2.83	2.25	2.83	0.94	8
10	8	16	2.83	2.50	2.83	0.89	8
11	8	16	2.83	2.75	2.83	0.85	8
12	8	16	2.83	3.00	3.00	0.87	16
13	8	16	2.83	3.25	3.25	0.90	16
14	8	16	2.83	3.50	3.50	0.94	16
15	8	16	2.83	3.75	3.75	0.97	16
16	16	32	4.00	2.83	4.00	1.00	16
17	16	32	4.00	3.01	4.00	0.97	16
18	16	32	4.00	3.18	4.00	0.94	16
19	16	32	4.00	3.36	4.00	0.92	16
20	16	32	4.00	3.54	4.00	0.89	16
21	16	32	4.00	3.71	4.00	0.87	16
22	16	32	4.00	3.89	4.00	0.85	16
23	16	32	4.00	4.07	4.07	0.85	32
24	16	32	4.00	4.24	4.24	0.87	32
25	16	32	4.00	4.42	4.42	0.88	32
26	16	32	4.00	4.50	4.50	0.90	32
27	16	32	4.00	4.77	4.77	0.92	32
28	16	32	4.00	4.95	4.95	0.94	32
29	16	32	4.00	5.13	5.13	0.95	32
30	16	32	4.00	5.30	5.30	0.97	32
31	16	32	4.00	5.48	5.48	0.98	32
32	32	64	5.66	4.00	5.66	1.00	32

12.0 TARGET SIZE ESTIMATION

Our main objective in developing MRSI was to find a means for SNR enhancement, and, at least for the example of Section 11.0, it appears that this has been achieved. Inspection of Figs. 28 and 29 shows that the SNR gain achieved by MRSI is quite close to that achieved by the theoretically optimum perfect synchronous integrator (PSI).

A second objective was to have MRSI develop an estimate for target size, and in this interest, the current algorithm clearly needs improvement. Inspection of Figs. 30 and 31 shows that the target size estimates developed by MRSI can easily be 30% in error for the previously discussed example. However, as discussed in this section, it appears straightforward to develop simple variants of the original algorithm that are capable of providing greatly improved estimates for target size. For example, we propose the following refined target size estimator, appropriate for the preceding one-dimensional example:

$$\hat{m} = Int\ [\tilde{m}]\ , \qquad (77)$$

where

$$log_2\ \tilde{m} = W\ log_2\ \hat{m}_1 + (1-W)\ log_2\ \hat{m}_2\ , (78)$$

and

$$W = -0.71 + 1.21\ [G(\hat{m}_1)/G(\hat{m}_2)]\ . \quad (79)$$

Quantity $\hat{m}_1$ in Eqs. 78 and 79 was previously defined by Eq. 73. Similarly, quantity $\hat{m}_2$ is defined implicitly by

$$G(\hat{m}_2) = \min_n G(n) = \min_n\ [n_o^{1/2}, m/(2n_o^{1/2})]\ .(80)$$

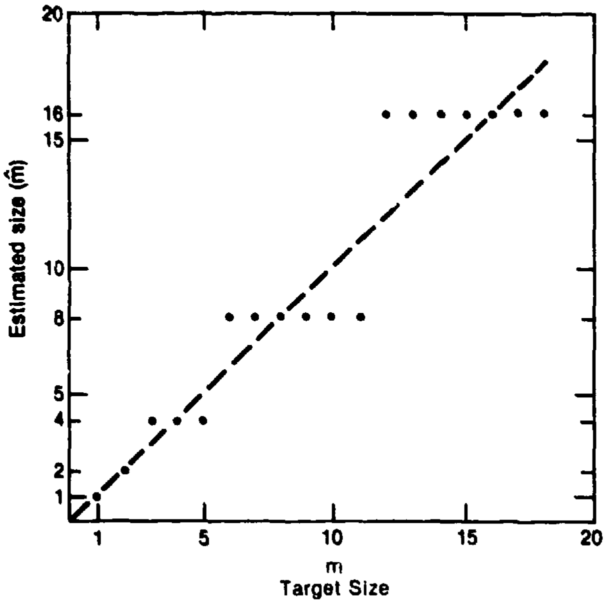


Figure 30. Initial estimate of target size, $\hat{m}_1$, from Table 4. Dashed line, $\hat{m}$ = m, denotes theoretically ideal performance, as achieved by the refined target size estimator, Equations (77)-(79), in the limit of high SNR.

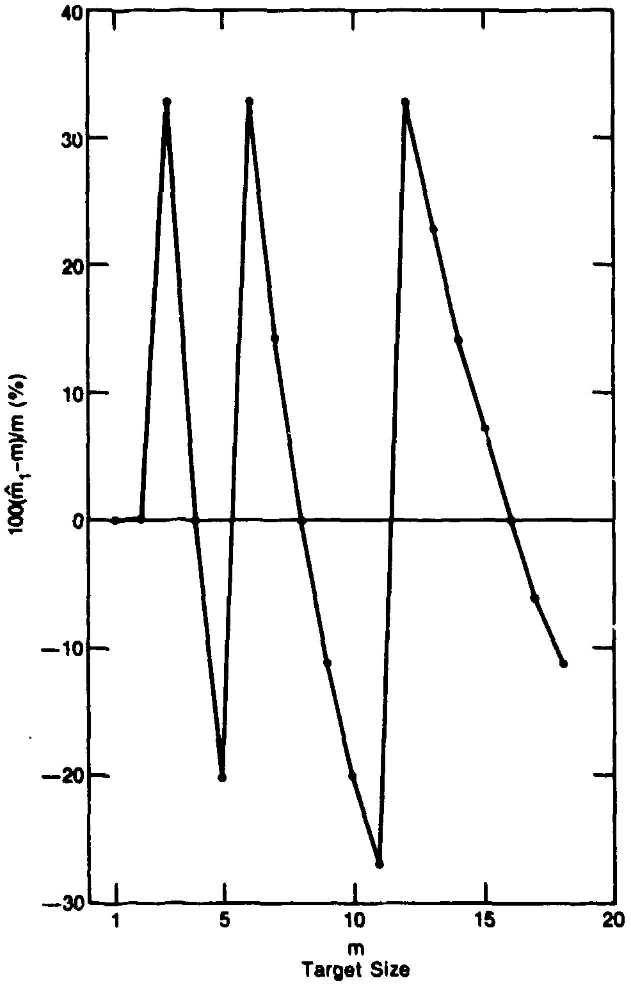


Figure 31. Size estimate percent error, derived from $\hat{m}_1$ values in Table 4, for the initial target size estimator.

The function *Int* [·] in Eq. 77 denotes rounding of its argument to the nearest integer value.

The form of Eq. 78 was suggested by the requirement that

$$log_2 \hat{m}_1 \geq log_2 \hat{m} > log \hat{m}_2 \ . \qquad (81)$$

The coefficients (-0.71, 1.21) in Eq. 79 were derived by imposing the dual requirements

$$\hat{m} = \hat{m}_1 \text{ , when } G(\hat{m}_1) = \sqrt{2}\, G(\hat{m}_2) \text{ , } \qquad (82)$$

and

$$W = ½ \text{ , when } G(\hat{m}_1) = G(\hat{m}_2) \ . \qquad (83)$$

Continuing in Table 5 the numerical example begun in Table 4, we find that the refined one-dimensional target size estimator provides in the limit of high SNR a perfect estimate of target size; i.e.,

$$\lim_{SNR \to \infty} \hat{m} = m \ . \qquad (84)$$

Although it would be straightforward to now explore via simulation the noise properties of our refined target size estimator, such as the size estimate variance, this has not yet been done.

The preceding one-dimensional size estimation algorithm, Eqs. 77 through 79, can be readily extended to the development of refined two-dimensional size estimates as well.

Table 5

Continuation of the numerical example begun in Table 4. Refined estimate of target size, $\hat{m}$, is developed via Eqs. 77 through 79. Comparison of left-most and right-most columns indicates that $\hat{m}$=m, i.e., target size is correctly estimated when signal-to-noise is high.

m	$\hat{m}_1$	$\hat{m}_2$	$\log_2\hat{m}_1$	$\log_2\hat{m}_2$	$G(\hat{m}_1)$	$G(\hat{m}_2)$	W	1-W	$\tilde{m}$	$\hat{m}$
1	1	2	0	1	1.00	0.71	1.00	0.00	1.00	1
2	2	4	1	2	1.41	1.00	1.C0	0.00	2.00	2
3	4	2	2	1	1.50	1.41	0.58	0.42	2.99	3
4	4	8	2	3	2.00	1.41	1.00	0.00	4.00	4
5	4	8	2	3	2.00	1.77	0.66	0.34	5.06	5
6	8	4	3	2	2.12	2.00	0.57	0.43	5.94	6
7	8	4	3	2	2.47	2.00	0.78	0.22	6.87	7
8	8	16	3	4	2.83	2.00	1.00	0.00	8.00	8
9	8	16	3	4	2.83	2.25	0.81	0.19	9.13	9
10	8	16	3	4	2.83	2.50	0.66	0.34	10.13	10
11	8	16	3	4	2.83	2.75	0.54	0.46	11.00	11
12	16	8	4	3	3.00	2.83	0.57	0.43	11.88	12
13	16	8	4	3	3.25	2.83	0.68	0.32	12.82	13
14	16	8	4	3	3.50	2.83	0.79	0.21	13.83	14
15	16	8	4	3	3.75	2.83	0.89	0.11	14.83	15
16	16	32	4	5	4.00	2.83	1.00	0.00	16.00	16
17	16	32	4	5	4.00	3.01	0.90	0.10	17.15	17
18	16	32	4	5	4.00	3.18	0.81	0.19	18.25	18

13.0 SHIP IMAGE DATA BASES

This section briefly describes the three ship image data bases that have been used in numerical experiments directed toward validating both the MRSI processor concept and a software implementation (App. A). The three sources of ship imagery used thus far include:

- NWC FLIR data,
- Mini FLIR data, and
- APL model silhouettes.

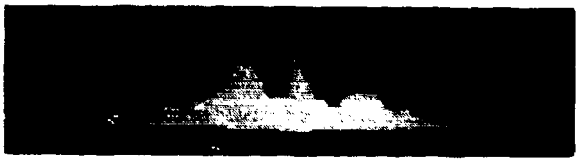


Figure 33. Measured ship thermal image thresholded to display what appear to be ac-coupling artifacts in front of and behind the ship, near the waterline.

The NWC data comprise 7537 thermal images of eight different ship classes, observed from the air at shallow depression angles, at aspects of 30° and 90° off-bow, for distances of from 6 to 20 nmi. As the data were obtained during a time of very good atmospheric visibility, the images consequently have very high SNR and are not directly suitable for realistically testing the limits of signal processing methods for SNR enhancement. For these data to be used, they were first degraded with progressively increasing levels of additive noise, as illustrated in Fig. 32.

Of course, much of the interest in working with these data, stemming from the "real world flavor" of measured imagery, is lost when the data are synthetically degraded. Since we had very little information about either the sensor or the data reduction process at the time this work was performed, and with what appears to be AC-coupling artifacts in some of the data (e.g., Fig. 33), we have made only limited use of these data thus far (cf. Section 14.1). It is our understanding that a second NWC ship image data set has been developed, containing images measured at lower SNR. Unfortunately, the low SNR NWC data were not available at the time this work was performed.

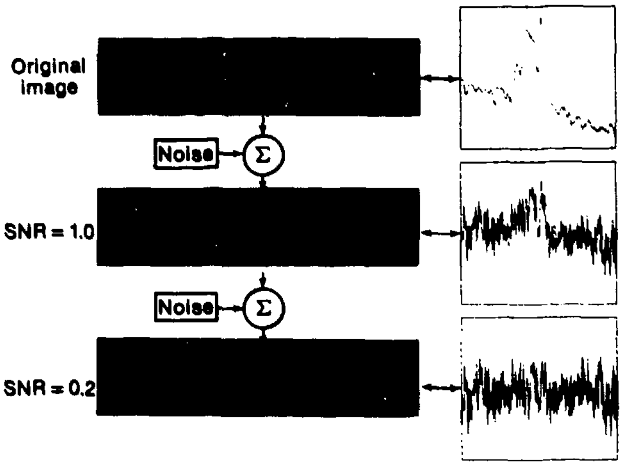


Figure 32. High SNR image (top) measured under unusually good atmosphere conditions is degraded with progressively increasing levels of additive noise to simulate observations obtained during less favorable weather. To the right of each image is shown a scan line obtained at the elevation indicated by a horizontal arrow.

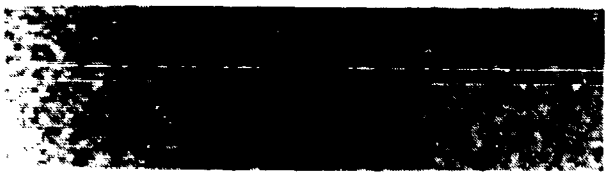


Figure 34a. Ship IR image digitized from miniFLIR analog video.

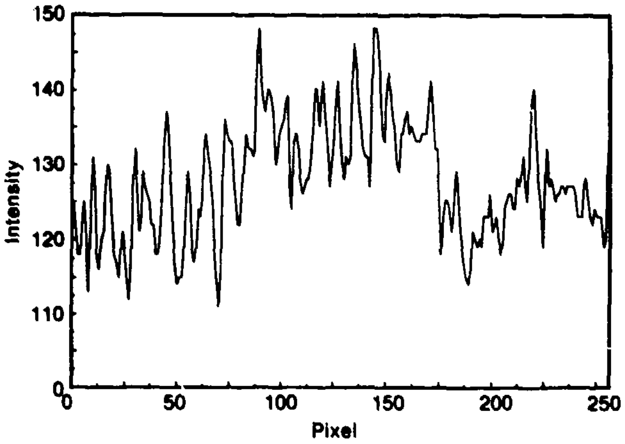


Figure 34b. Scan line through the peak intensity pixel in the miniFLIR image shown in Part a.

A second source of ship thermal imagery was some mini FLIR analog video that became available via a previous APL program. Several frames of this data were digitized by K. Constantikes (APL/FIF) for our use. Since a number of important sensor characteristics and the environmental conditions prevailing at the time of measurement were unfortunately not known, we have made only limited use of the mini FLIR data (cf. Section 14.1). Visual inspection of the data, both analog and digital, shows the SNR to be only marginally adequate for visual detection (cf. Fig. 34). These data are therefore of at least qualitative interest, since, once digitized, they can be used to exercise low SNR detection processing without first requiring synthetic noise insertion.

Our third source of ship image data, the APL model silhouette data base, was originally developed in connection with an earlier APL Independent Research and Development (IRAD) effort [5,28]. The data were acquired by digitizing TV images of scale models of five ship types at 21 distinct aspects around the starboard bow quadrant for each of three simulated ranges (cf. Fig. 35). Thus, a total of 315 silhouette images is available (5 ship types × 3 ranges × 21 aspects). The models were cast in lead at a 1:1250 scale, and are representative of ships at the waterline. The images are noise-free, black/white silhouettes, digitized to 512 × 512 pixels per image.

A wide variety of effects, some or all of which may figure importantly in measured imagery, are obviously not present in the model silhouette data, viz.,

- background structure (e.g., due to waves, reflected clouds, natural ocean thermal variations, and slicks [26]),
- ship thermal structure (e.g., stack hot spots),
- atmospherics,
- ship's reflection in the water,
- ship's wake, and
- assorted sensor artifacts (e.g., AC-coupling, "striping" due to uncompensated detector nonuniformities, vignetting, etc.).

One consequence of these limitations, for example, is that the model silhouette data cannot be used to assess the potential usefulness of thermal gray scale information in assisting classification.

Certain real-world characteristics can be imparted to the model silhouettes via modeling. For example, Fig. 36 illustrates how model silhouettes (three images at figure upper left) may be combined with ship IR signature data (curve at figure upper right) that incorporate both ship thermal modeling [20] and a statistical treatment of weather effects [12]. Seeker sensitivity is described by the NEΔT parameter (curve at figure lower right). Output from the image model is a set of images (figure lower left) having SNRs that reflect a wide variety of phenomenological considerations.

It is pointed out in [5] that the model silhouettes appear well-suited to developing ship recognition techniques based on ship profile information, since the ship profile may be substantially the same for IR imagery as for TV imagery. Of greater interest to the present application, it is noted that the well-controlled measurement conditions for the model silhouettes facilitate the development of curve fits for ship projected area Vs. viewing aspect. It is found that, with relatively small error (cf. Fig. 37)

$$A_{ship}(\theta) \cong \begin{matrix} 4750 \, sin \, \theta + 645 \, cos \, \theta \ , & carrier \\ 880 \, sin \, \theta + 90 \, cos \, \theta \ , & frigate \end{matrix} \ , \quad (85)$$

over the range 7° < θ° ≤ 90° (beam aspect), and

$$A_{ship} \, (\theta) \cong A_{ship}(7°) \quad , \quad 0° \leq \theta \leq 7° \ . \quad (86)$$

It follows from Eq. 85 that the processor SNR model, Eq. 70, can now be written as

$$G(\theta,R) \cong \begin{matrix} (37.2\eta/R_{u}) \cdot [\,(sin\theta + 0.14 \, cos\theta)/\alpha\beta]^{1/2}, \ carrier \\ (16\eta/R_{o}) \cdot [\,(sin\theta + 0.10 \, cos\theta)/\alpha\beta]^{1/2}, \ frigate \end{matrix} \ ,$$

(87)

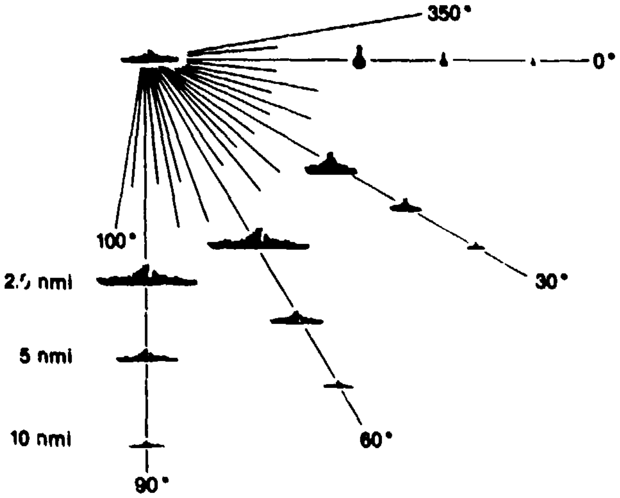


Figure 35. APL model silhouettes are available for 5 ship types, each observed at 21 aspect angles and 3 ranges [28].

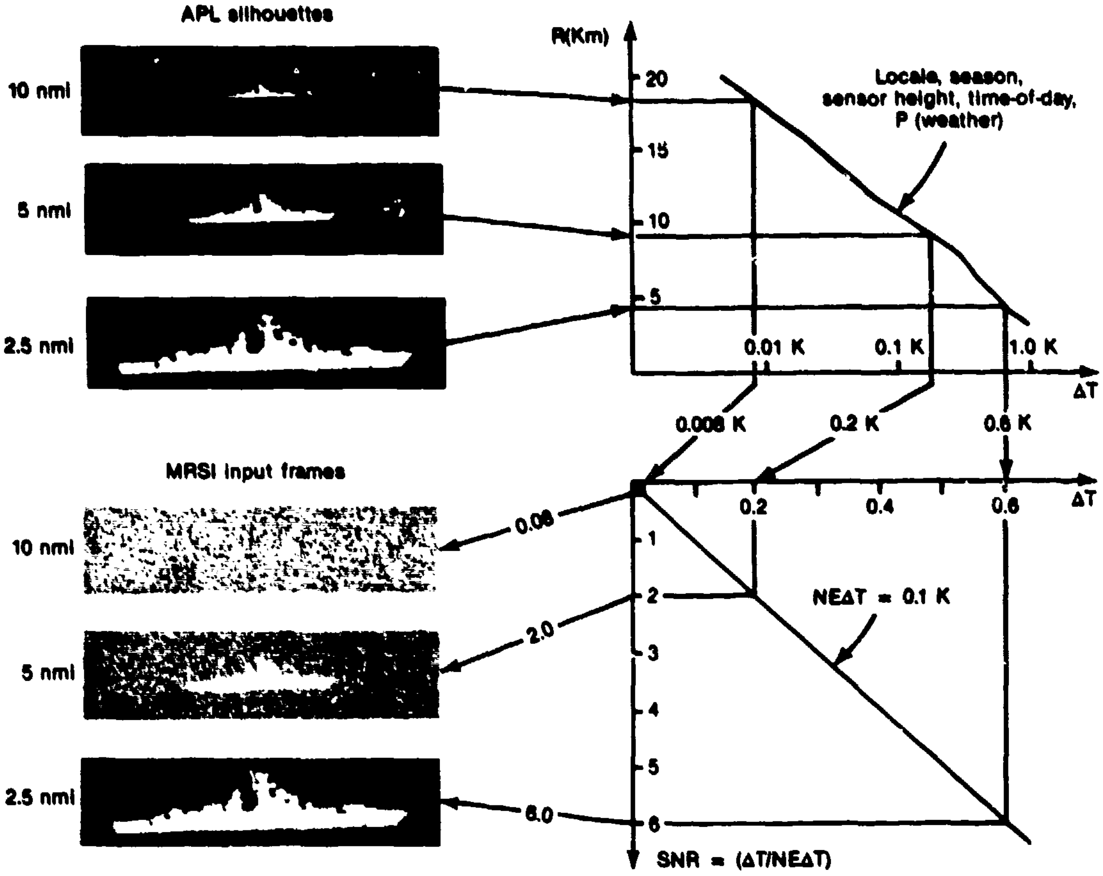


Figure 36. Image model combines idealized silhouettes (3 images, upper left) and ship IR signature data. The IR signature (curve, upper right) incorporates both ship thermal modeling and a statistical treatment of weather effects [12].

over the range 7° < $\theta \leq$ 90°, and, at smaller aspects

$$G(\theta,R_o) \cong G(7°, R_o) \ , \quad 0° \leq \theta \leq 7° \ . \quad (88)$$

We note that R_o in Eq. 87 is range in nmi units, and (α, β) are in mr units.

For illustrative purposes, we plot in Fig. 38 the variation of processing gain with range, as obtained from Eq. 87, for two aspects (90° and 45°), against a frigate target. In obtaining Fig. 38, a spatial resolution of $\alpha = \beta = 0.15$ mr and a processor efficiency of $\eta = 1$ are assumed.

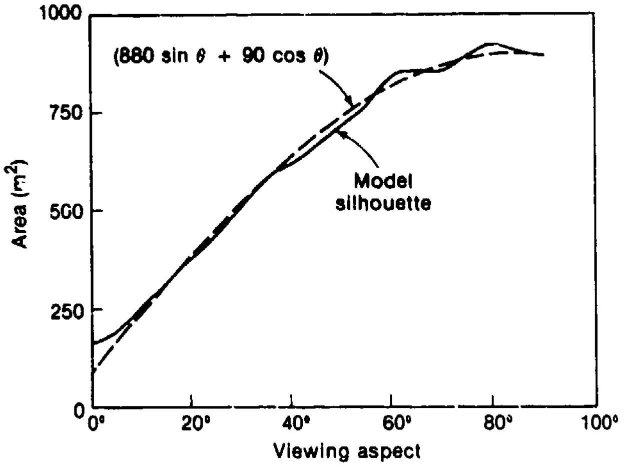


Figure 37. Ship projected area vs. viewing angle obtained from APL model silhouette imagery of a frigate (solid line). Curve fit (dashed line) was obtained heuristically.

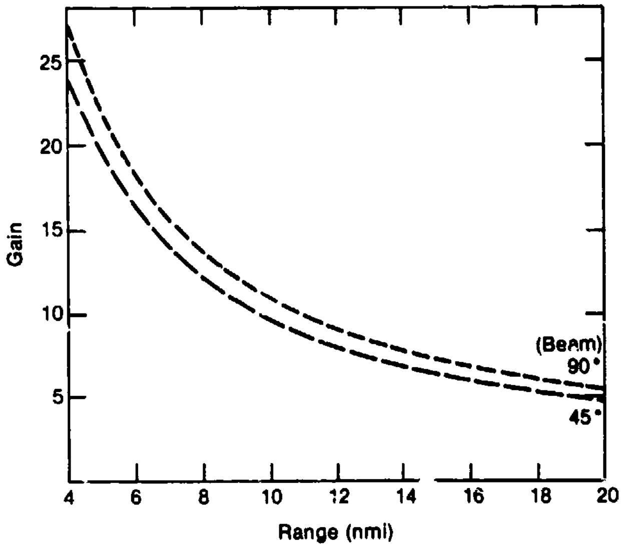


Figure 38. Processing gain (G) vs. range, against frigate target, from Eq. (87). Assumed spatial resolution is $\alpha = \beta = 0.15$ mr; assumed processor efficiency is $\eta = 1$.

14.0 RESULTS OF IMAGE-BASED SIMULATIONS

An image-based simulation of the MRSI processor has been coded in FORTRAN (App. A). The simulation operates on images of 64 x 256 pixels, at 35 spatial resolutions. The program runs, with minor changes, on both the APL NAS mainframe and on an IBM PC AT computer and requires about 450 kbytes of memory. The listing provided in App. A is for the IBM PC AT version of the code. All simulation results provided in this section (the results of more than 2800 processed images) were developed on an IBM PC AT.

14.1 INITIAL QUALITATIVE RESULTS

Our first MRSI simulations against ship imagery were performed using NWC imagery, synthetically degraded in SNR, as discussed in connection with Fig. 32. It was observed for these first simulations that correct detections and reasonably accurate size estimates were developed with input SNR values of 0.2. One of the results of these early efforts is shown in Fig. 39, in which a ship at 13.5 nmi range is correctly detected and sized, for $SNR_C = 0.2$.

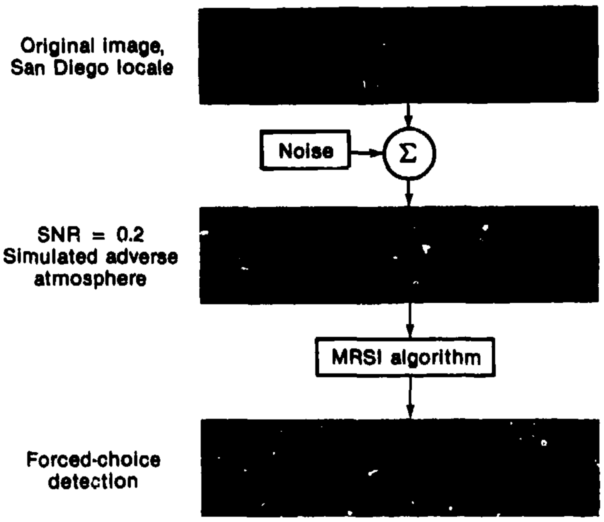

Original image, San Diego locale
Noise
Σ
SNR = 0.2 Simulated adverse atmosphere
MRSI algorithm
Forced-choice detection

Figure 39. Initial qualitative evidence of correct simulation performance. Original high-SNR NWC ship image (top) was degraded to SNR = 0.2 (center) to simulate observation in a less favorable atmosphere. Degraded image was input to FORTRAN simulation (App. A) and the ship subsequently detected and sized (bottom). Range to ship is 13.5 nmi.

A similar qualitative result was obtained by exercising MRSI against a mini FLIR image of uncertain SNR (cf. Fig. 34). As shown in Fig. 40, in this case, we ran MRSI in an experimental multi-pass configuration, and, after three passes, were provided by MRSI with a highly ship-like composite. The ship in this data was observed at a range of about 10 nmi.

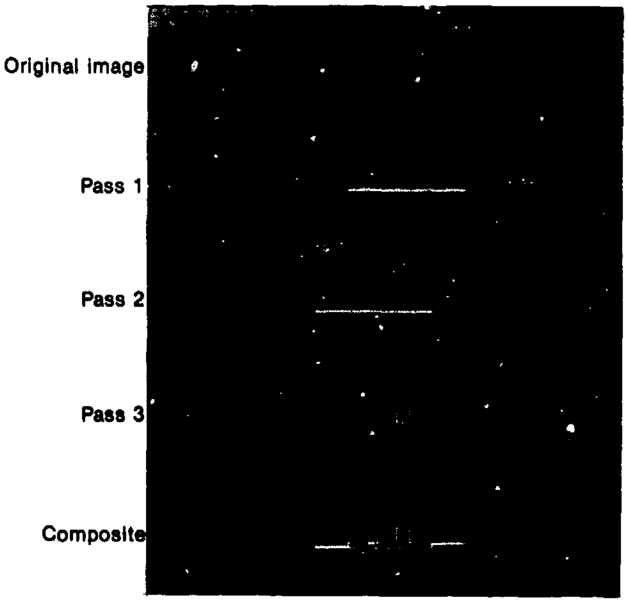


Figure 40. Multipass detection of low contrast miniFLIR image provides shape information. Original image shown at top; composite product of 3-pass detection shown at bottom. Range to ship is 10 nmi.

14.2 PROCESSOR EFFICIENCY

The input images for the simulations described in this section were various APL model silhouettes (cf. Section 13.0), to which was added a small amount of Gaussian noise to obtain

$$SNR_C = 5 \ . \qquad (89)$$

As discussed in connection with Fig. 26, the detected targets are characterized by MRSI in .erms of five parameters, viz.,

- position (i_0, j_0),
- size (k_0, n_0), and
- contrast C_0.

In addition, as discussed in connection with Fig. 26 and Eq. 67, the processor develops an estimate of the noise variance, σ_{k_n}, for each spatial filter output. An estimate for the detection SNR, defined in Eq. 12, is then obtained from the results of simulation as

$$\widehat{SNR}_{det} = C_o/\sigma_{k_o n_o} \ . \qquad (90)$$

However, from Eq. 13

$$\eta = G \cdot N_{PIX}^{-1/2} = (SNR_{det}/SNR_C) \cdot N_{PIX}^{-1/2} \ . \ (91)$$

We obtain from Eqs. 90 and 91 the following estimate, η_1, for processor efficiency

$$\hat{\eta}_1 = (\widehat{SNR}_{det}/SNR_C) \cdot N_{PIX}^{-1/2} = \hat{G} \cdot N_{PIX}^{-1/2}, \qquad (92)$$

where SNR_C (Eq. 89) and N_{PIX} are known a priori, and $\widehat{SNR}_{det}$ is an output of the simulation.

Since Eq. 90 provides only an estimate for SNR_{det}, we improve the estimate by performing each simulated detection a number of times (' ach time with a different random number seed) and then averaging over the resulting individual SNR_{det} estimates.

Alternatively, Eq. 71 and the position and size estimates provided by the simulation, (i_o, j_o) and (k_o, n_o), can be used to estimate η as

$$\hat{\eta}_2 = A/(k_o n_o \cdot N_{PIX})^{1/2} \ . \qquad (93)$$

Calculations of processor efficiency have been performed for 13 model silhouettes (Table 6). The first series of Monte Carlo simulations was performed with a very large number (120) of trials to obtain an estimate for how many Monte Carlo trials were needed to achieve adequate convergence of the estimate for SNR_{det}. The silhouette image used in all 120 trials was a frigate profile, for range = 10 nmi, aspect = 90°; the only differentiating aspect of the 120 input images was in the random number seed used in generating the additive noise. The results of these 120 trials are given in Fig. 41 as a plot of

$$\hat{G} = \widehat{SNR}_{det}/5 \ ,$$

as a function of the number of Monte Carlo trials.

Table 6
Calculations of processor efficiency η have been performed for 13 model silhouettes.

Ship	Aspect (°)	Ranges (nmi)
Frigate	10	5, 10
	45	5, 10, 15
	90	10, 20
Carrier	10	10, 20
	45	20, 30
	90	20, 30

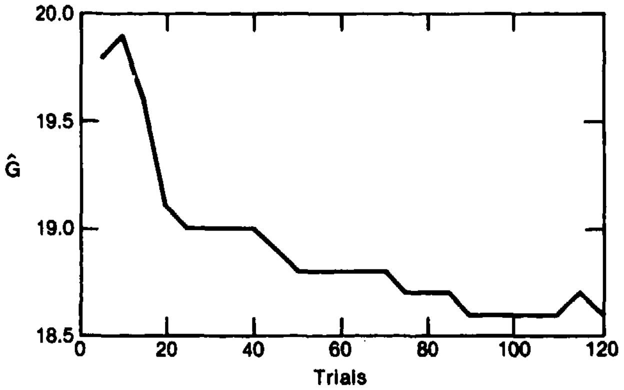


Figure 41. Estimated processing gain, as a function of the number of Monte Carlo trials. The same silhouette image was used in all 120 trials (frigate at 10 nmi, aspect = 90°). 120 distinct images were created by adding to the noise-free ship image 120 different noise images.

We surmised from this numerical experiment that no more than 15 Monte Carlo trails are required to develop an estimate for $\hat{G}$ accurate to within 5 or 10% of the fully converged value. All consequent determinations of η via simulation for the remaining 12 ship profiles listed in Table 6 were established as averages over 15 Monte Carlo trials.

Estimates of processor efficiency for the 13 ship profiles, obtained using Eq. 92, are provided in Table 7. Estimates of η based on Eq. 93 are given in Table 8.

The dispersion of $\hat{\eta}_1$ values in Table 7 is bounded by

$$|\ \hat{\eta}_1 - 0.82\ | < 0.08 \ , \qquad (94)$$

with mean and variance

$$E(\hat{\eta}_1) = 0.81 \qquad (95)$$
$$Var(\hat{\eta}_1) = (0.048)^2 \ .$$

The dispersion of η_2 values in Table 8 is bounded by

$$|\ \hat{\eta}_2 - 0.76\ | < 0.05 \ , \qquad (96)$$

Table 7
Processor efficiency, η, is estimated using Eq. 92 and results from image-based simulations, for 13 ship silhouettes. Each value of $\hat{G}$ in this table is established as an average over 15 Monte Carlo iterations; thus, a total of 195 images was processed to obtain these results. Assumed IFOV = 0.073 mr

Ship	Aspect (°)	Ranges (nmi)	N_{PIX}	$\hat{G}$	$\hat{\eta}_1$
Frigate	10	5	479	16.77	0.77
		10	139	9.32	0.79
	45	5	1247	30.69	0.87
		10	355	15.66	0.83
		15	139	9.49	0.80
	90	10	477	19.63	0.90
		20	122	8.90	0.80
Carrier	10	10	702	20.66	0.78
		20	175	10.00	0.76
	45	20	434	18.14	0.87
		30	182	10.66	0.79
	90	20	555	20.24	0.86
		30	248	11.82	0.75

Table 8
Processor efficiency for 13 ship profiles as estimated from Eq. 93

			G(PSI)	G(ISI)	
Ship	Aspect (°)	Ranges (nmi)	$N_{PIX}^{1/2}$	$A/(k_o\, n_o)^{1/2}$	$\hat{\eta}_2$
Frigate	10	5	21.89	16.62	0.759
		10	11.79	9.19	0.780
	45	5	35.31	26.66	0.755
		10	18.84	13.74	0.730
		15	11.79	9.19	0.780
	90	10	21.84	17.50	0.801
		20	11.05	8.75	0.792
Carrier	10	10	26.05	19.49	0.748
		20	13.23	9.72	0.735
	45	20	20.83	16.18	0.777
		30	13.49	10.25	0.760
	90	20	23.56	18.03	0.765
		30	15.75	11.23	0.713

with mean and variance

$$E(\hat{\eta}_2) = 0.76 \qquad (97)$$
$$Var(\hat{\eta}_2) = (0.025)^2 \; .$$

Based on a total of 195 Monte Carlo simulations with 13 different ship profiles, the efficiency of MRSI can be characterized as

$$\eta \cong 0.8 \; , \qquad (98)$$

accurate to about ±10%, for frigate and aircraft carrier profiles, observed at ranges from 5 to 30 nmi, and for aspect angles from 10° to 90°.

The rather remarkable invariance of η with respect to ship type, scale, and viewing aspect was perhaps presaged by our earlier analytically derived results for bar targets, Fig. 29. In a speculative vein, we note that the bar target results can be summarized as

$$| \eta - 0.92 | < 0.08 \ .$$

Heuristically extending the bar target results to two dimensions

$$\hat{\eta} \cong (0.92)^2 \cong 0.85 \ ,$$

we get a result fairly close to Eq. 98.

14.3 THRESHOLD SNR FOR FORCED CHOICE DETECTION

We explored systematically the variation of probability of forced choice detection (P_D) with SNR for three ship profiles, involving 2500 Monte Carlo trials (Table 9). For each ship profile, for each SNR value, 100 Monte Carlo trials were performed; the relative number of times the declared target overlapped the true target was used as an estimate for P_D.

Our results are presented in Fig. 42 as plots of P_D vs. SNR, parametric in viewing aspect. Each of the 25 data points in Fig. 42 was established via 100 image-based simulations.

The data are replotted in Fig. 43 as P_D vs. SNR_{det}, where SNR_{det} is given by Eq. 12 as

$$SNR_{det} = \eta \cdot SNR_C \cdot N_{PIX}^{1/2}$$

with an assumed value of $\eta = 0.81$. Also shown as a solid line in Fig. 43 is a plot of the function

$$P_D(SNR_{det}) = \Phi(SNR_{det} - 2.8) \ , \qquad (99)$$

where

$$\Phi(t) = (1/\sqrt{2\pi}) \int_{-\infty}^{t} exp(-t^2/2)dt \ .$$

Interestingly, it has been found in psychovisual experiments performed with human observers of electronic displays that a 50% probability of detection against rectangular target images requires a "display SNR" of 2.8 [4, p. 86]. Equation 99 is, as per the discussion in [3, p. 197], a curve fit to experimental data descriptive of human vision system (HVS) performance.

The fact that the P_D data for all three aspect angles are nicely fit in Fig. 43 by a single universal curve suggests that P_D statistics may be derived from the universal curve for a wide variety of ship target profiles. Testing the generality of this hypothesis would, however, clearly require more simulation and analysis than presented in this report.

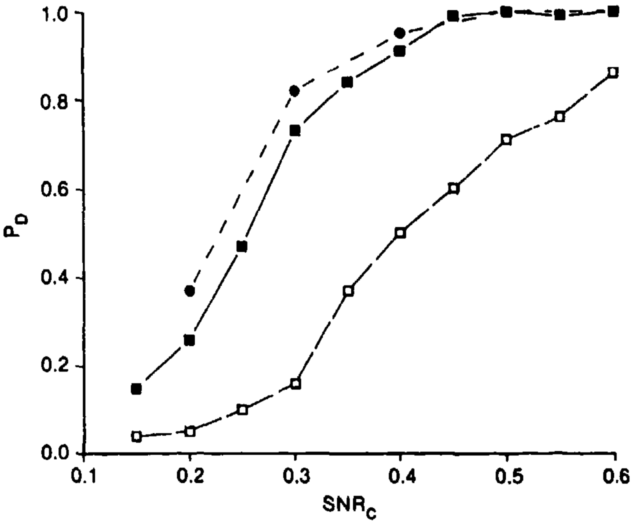


Figure 42. Probability of detection (P_D) vs. channel SNR (SNR_C). Frigate silhouette having range resolution product = 0.73 nmi-mr. Curves are parametric in viewing aspect: ● = 90°, ■ = 45°, □ = 10°.

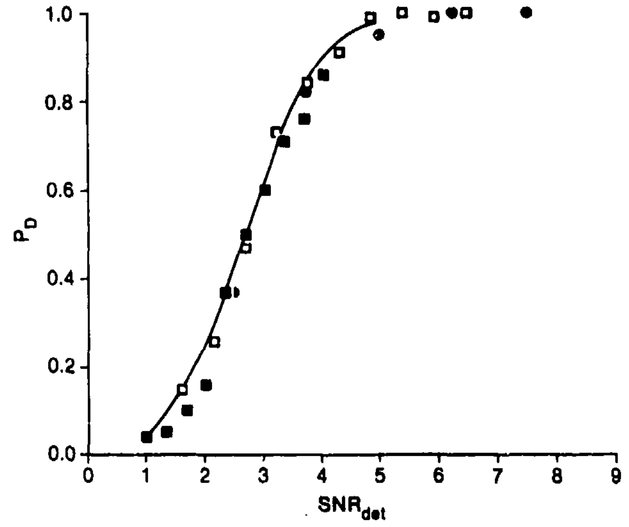


Figure 43. Detection probability (P_D) vs. detection SNR (SNR_{det}, defined in Eq. 12). Frigate silhouette, (range x resolution) = 0.73 nmi-mr. Three symbol types correspond to different viewing aspects: ● = 90°, □ = 45°, ■ = 10°. Solid line is an analytical curve fit to experimental psychovisual data, Eq. (99).

Table 9
Probability of detection statistics were accumulated as a function of SNR for a total of 2500 Monte Carlo trials. Range · resolution product = 0.73 nmi · mr

Ship	Aspect (°)	N_{PIX}	Range of SNR Values	SNR Increment	Monte Carlo Trials Per SNR Value
Frigate	10	139	0.15-0.60	0.05	100
	45	355	0.15-0.60	0.05	100
	90	477	0.20-0.60	0.10	100

15.0 SYSTEM CONCEPTS FOR CUED DETECTION AND ATR

As discussed in Section 1.0 the current study was motivated by an interest in the antiship missile (ASM) application. In this section, we discuss some top-level ASM system concepts in which the MRSI processing approach could be employed.

Figure 44 depicts a tactical scenario in which an ASM has been launched in the general direction of a hostile surface force, with the problem of target selection to be addressed post-launch. The missile seeker acquires a first candidate ship target, which is subsequently classified as ineligible for attack. The ASM continues its flight, and, as depicted in Fig. 44, acquires a second ship, which it classifies and engages as a high-priority hostile combatant.

The initial pre-acquisition surveillance performed by the ASM seeker is presumably performed over a field-of-view much larger than that of typical thermal imaging systems and TVs. The initial target acquisition could be performed by MRSI processing, which then directs the seeker's further attention to the neighborhood of "objects" requiring classification. Thus, the MRSI processor is followed by a classification processor that makes a series of judgments regarding the nature of each object, viz., ship/non-ship; combatant/non-combatant; hostile/friendly; high value/low value; etc. Two system concepts for classification will be described.

Figure 44 shows a system concept in which video is telemetered from the ASM to an aircraft, where target classification is performed and attack sequencing initiated by a human operator. According to this concept, MRSI plays the role of a target "cuer", reducing the bandwidth requirements of the video link and unburdening the human image interpreter from the need to perform visual search over a wide field-of-view.

Figure 45 depicts an alternative system concept in which classification is performed on-board the missile by an autonomous target recognition (ATR) algorithm.

A potential problem with both system concepts is that the seeker's acquisition range is likely to be much greater than the range at which accurate classification can be performed, since classification both by eye and by ATR is likely to have far more demanding SNR requirements than the MRSI acquisition algorithm. Although the ASM could simply fly toward each potential target until the SNR grows large enough to permit classification, this approach could result in the waste of considerable fuel capacity.

A second, more satisfactory, solution to the acquisition/classification SNR mismatch problem has been suggested by W. J. Tropf (APL). Following MRSI acquisition, the seeker could be directed into a "classification mode", in which scan is performed very slowly over a very narrow field-of-view in the neighborhood of the potential target. The reduced angular rate-of-scan, combined with a proportionally reduced post-detector-amplifier electrical bandwidth, then provides the SNR boost required for classification, without sacrificing spatial resolution.

For example, the classification mode may be per-

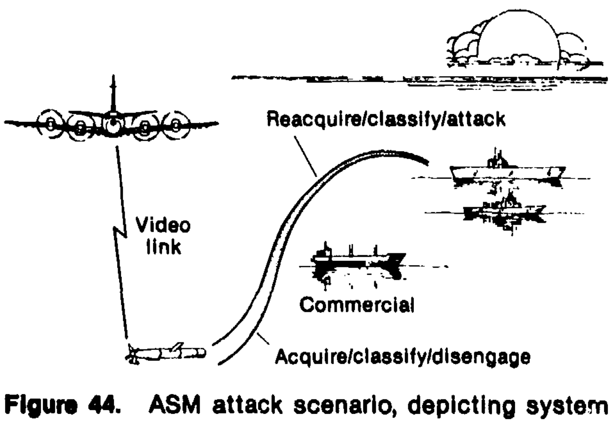


Figure 44. ASM attack scenario, depicting system concept in which target acquisition is performed by MRSI, and target classification is performed off-board by a person inspecting cued images transmitted via video link from the ASM.

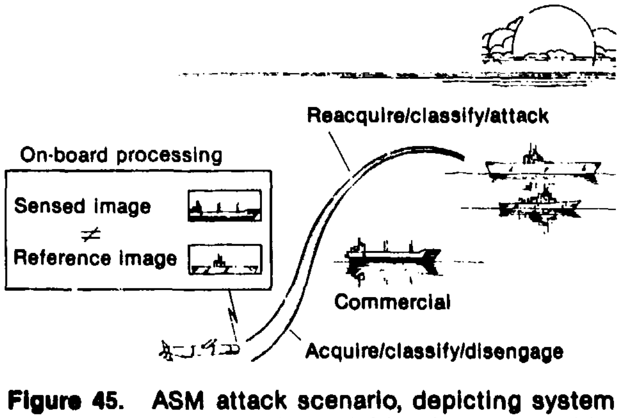


Figure 45. ASM attack scenario, depicting system concept in which target acquisition is performed by MRSI and target classification is performed by on-board ATR.

formed with an angular rate-of-scan 25 times slower than the acquisition mode,

$$\dot{\theta}_C = \dot{\theta}/25 \ ,$$

where $\dot{\theta}$ was defined previously in connection with Eqs. 25 and 27. If the detected waveforms are sampled at the same rate during slow scan (classification mode) as during fast scan (acquisition mode), the data during slow scan will be over-sampled 25-fold, from the standpoint of satisfying Nyquist's criterion and preserving spatial detail. Consequently, the A/D process can be followed by a pre-classification processor configured as per Fig. 8 and Eq. 23, in which the data are smoothed and decimated-in-time 25-fold, with a consequent 5-fold SNR gain.

The cued slow-scan classification mode is potentially useful to both system concepts, Figs. 44 and 45, i.e., regardless of whether classification is performed by ATR or by a human.

16.0 A NOVEL METHOD FOR CLOUD CLUTTER SUPPRESSION

Cloud reflections in the water are the most intense natural source of sea surface thermal structure [26], and are consequently expected to offer the most severe source of false alarms to this (and other) ship IR detection schemes. In this section we present a novel method for suppressing false alarms induced by cloud reflections from the sea.

Our new clutter suppression approach entails corelating MRSI detections obtained below the horizon (due either to a ship or a cloud reflection) with detections obtained above the horizon at the same azimuth: the presence of cloud structure above the candidate target's location indicates a likely false target, while the absence of clouds indicates a likely true target.

We illustrate the method with the aid of Fig. 46, depicting the spatial variation of sky radiation temperature (top) and sea surface radiometric temperature (bottom), as measured at 10.6 μm, in the presence of clouds. These data were measured by JHU/APL with a calibrated dual-band IR radiometer installed in a Navy P-3 aircraft [26]. For example, the sea surface thermal artifact labelled A′ in the lower trace is clearly indicated as a cloud reflection, since it correlates directly with thermal structure observed above the horizon, i.e., the artifact labelled A in the upper trace of the figure.

The vector of attributes attached to each MRSI detection (cf. Fig. 26) can be augmented with a "quality" measure derived from correlation with the sky measurements.

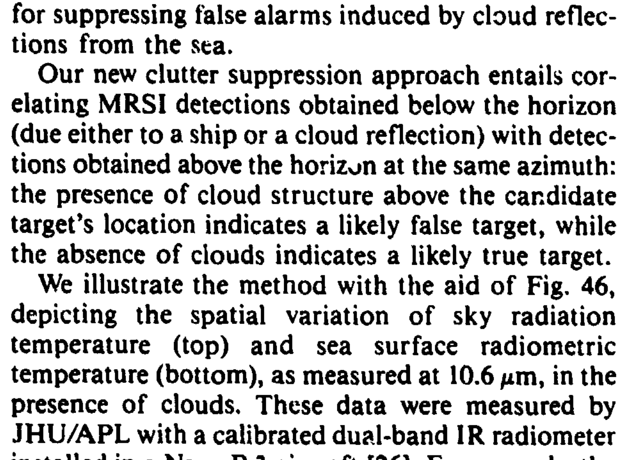


Figure 46. Spatial variation of sky radiometric temperature (top) and sea surface temperature (bottom) measured by APL at 10.6 μm, in the presence of clouds [26]. Sea surface artifacts labelled with primed letters in the lower trace are reflections of cloud structures labelled with corresponding unprimed letters in the upper trace.

The mechanization of our cloud clutter discriminant is simplified by making use of the fact that a cloud's contrast against the sky is generally several hundred times greater than the contrast of the cloud's reflection against the sea. For example, we see from Fig. 46 that cloud/sky contrast is about 50 K, while the reflection/sea contrast is only about 0.1 K. In addition, since there is no particular benefit to be derived from spatially resolving the fine details within the cloud structure, the sky radiance data may be acquired by means having greatly reduced resolution relative to the means used for detecting ships against the ocean background. In Table 10 we compare approximate sensitivity and spatial resolution requirements for cloud detection with corresponding requirements for ship detection.

As a first implementation possibility, we consider obtaining sea/ship radiance data on a forward scan in azimuth, and sky/cloud radiance data on the backscan. In this case, the backscan is stepped upwards in elevation, and executed at a greatly increased angular rate-of-scan, relative to the forward scan. Considerable image smear is tolerable on the backscan, due to the relaxed spatial resolution requirements for the sky measurements (cf. Table 10). Moreover, if necessary, excessive smear can be prevented simply by widening the post-detector electronic passband; the consequent increase in noise is accomodated by the relaxed sensitivity requirements during the backscan (cf. Table 10). Relaxed spatial resolution requirements in elevation can be exploited as a reduction in signal processing requirements during the backscan. This is accomplished by adding together the detector outputs in groups (say, of five each), to obtain a single reduced-resolution video channel for each group.

Table 10
Sensitivity and spatial resolution requirements for ship detection and cloud/clutter detection (approximate values).

	Ship Detection	Cloud Detection
Field of Search	Below horizon	Above horizon
Sensitivity, NEΔT	0.1 K	5 K
Spatial Resolution	0.2 mr, square	1 mr, square

A second implementation possibility is to perform the sea and sky measurements simultaneously (i.e., on the same scan) by having a partitioned focal plane: small elements at the bottom of the focal plane are used for below-horizon search, and much larger detectors at the top of the focal plane are used for above-horizon search. The optics design for this sensor concept is complicated by the requirement for wide elevation instantaneous field-of-view. However, this problem is partially ameliorated by the fact that a considerable degradation in the optics blur can be tolerated over the upper part of the focal plane.

Yet a third implementation possibility is to perform the sky search with a separate telescope. The telescope used for sky search can be very small, perhaps just 0.5-1 inch in aperture.

Finally, additional sensor concepts for nearly simultaneous sky/sea search can be based on cross-horizon scanning (discussed briefly in connection with Eqs. 60-62). The principal penalty imposed by this approach is a relatively complicated scanning mechanism, compared to seeker heads that scan in azimuth.

17.0 CONCLUSIONS

A signal processing algorithm (MRSI) has been devised to maximize the detection range of imaging IR seekers against ship targets. The algorithm comprises a bank of two-dimensional spatial filters, having rectangular kernels matched to targets of various aspect ratios and sizes, implemented in a separable form that appears well suited to processing in real-time image data acquired sequentially by column. The performance of MRSI has been studied analytically for simple bar-target-in-noise images and studied numerically via simulations performed on a number of measured and simulated ship images.

The SNR gain provided by MRSI (relative to hot-spot detection) can be estimated as

$$\eta N_{PIX}^{1/2}$$

where N_{PIX} is the number of pixels occupied by the target in the image, and $\eta = 0.81$, ±10%. The processor efficiency, η, is remarkably invariant with respect to sensor/ship range, viewing aspect, and ship class.

The principal simulation results are plots of detection probability (P_D) vs. SNR (Figs. 42 and 43), based on 2500 image-based Monte Carlo trials. A curve fit to the numerically-developed probabilities of detection appears to provide a universal curve that can be used predictively for ships of differing class, observed at a variety of aspects and distances. Interestingly, the curve fit to MRSI/Monte Carlo P_D results coincides with a previously published curve fit to data obtained from visual detection experiments performed with human observers of TV displays [3].

Section 15.0 describes two ASM system concepts, distinguished by the means of target classification, viz., classification performed on-board the missile by an ATR algorithm or off-board by a person inspecting imagery telemetered from the missile. Using MRSI to cue a slow-scan/high sensitivity "classification mode" results in substantial predicted improvements for both system concepts.

In Section 16.0 we present a new method for suppressing false alarms caused by cloud reflections from the sea surface.

ACKNOWLEDGMENTS

Many of the ideas contained in this report were originally developed with APL IRAD support. Thanks are due to B. G. Boone (APL/F1F) for encouraging continuation of this work beyond the original IRAD effort. Thanks also are due to W. J. Tropf (APL/F1F) for his technical review of an early version of this work. The idea of using MRSI to cue a slow-scan classification mode, discussed in Section 15.0, is due to W. J. Tropf. We thank K. Constantikes (APL/F1F) for digitizing several mini FLIR frames for our use. The silhouette image data base developed under an earlier APL IRAD program through the efforts of D. K. White (APL/F1E) and F. W. Riedel (APL/F1E) was extremely useful to us. We thank R. F. Gasparovic (APL/STR) for providing Fig. 46.

REFERENCES

1. Paul F. Walker, "Smart Weapons in Naval Warfare", Scientific American, Vol. 248, No. 5, May 1983, pp. 53-61.

2. B. G. Boone, F. W. Riedel, and J. F. Walter, "Survey of Target Recognition Technology", JHU/APL FS-80-147, July 1980.

3. F. A. Rosell and R. H. Willson, "Recent Psychophysical Experiments and the Display Signal-to-Noise Ratio Concept", Chapter 5 in Perception of Displayed Information, Lucien M. Biberman, ed., Plenum Press, New York, 1973.

4. F. Rosell and G. Harvey, "The Fundamentals of Thermal Imaging Systems", Naval Research Laboratory, Washington, D.C., NRL Report 8311, May 10, 1979.

5. F. W. Riedel and D. K. White, Jr., "A Comparison of Pattern Recognition Discriminant Sets for Autonomous Ship Target Classification", JHU/APL FS-83-202, September 1983.

6. A. F. Milton, G. L. Harvey, and A. W. Schmidt, "Comparison of the 3-5 Micrometer and 8-12 Micrometer Regions for Advanced Thermal Imaging Systems: LOWTRAN Revisited", Naval Research Laboratory, Washington, D.C., NRL Report 8172, 30 Dec. 1977.

7. Stanley T. Smith, "3 to 5 Versus 8 to 12 Micrometers Spectral Band Selection for Anti-Ship Sensors (Level 3)", NWC Technical Memorandum 5083, Naval Weapons Center, China Lake, CA, June 1983.

8. Optical Engineering, March 1987, Vol. 26, No. 3, Special Issue on Focal Plane Arrays.

9. "Optical and Infrared Detectors", R. J. Keyes, ed., Topics in Applied Physics, Vol. 19, Springer-Verlag, New York (1977).

10. Robin N. Strickland and M. R. Gerber, "Estimation of Ship Profiles from a Time Sequence of Forward-Looking Infrared Images", Optical Engineering, Vol. 25, No. 8, Aug. 1986, pp. 995-1000.

11. R. A. Steinberg, "Calculated Acquisition Ranges of IR Anti-Ship Seekers", JHU/APL F1F(1)86-U-060, 19 March 1986.

12. D. M. Wilson, "Ship Infrared Signature Predictions, Vol. 1: Introduction and 8-12 μm Contrast Temperature Climatologies", Naval Surface Weapons Center, Silver Spring, MD., NSWC TR84-510, May 1, 1985.

13. James R. Hamer and Richard Satterfield, "Performance Comparison of a Laboratory Digital Tracker to the Human Eye", in Proceedings of the Open Sessions of the Workshop on Imaging Trackers and Autonomous Acquisition Applications for Missile Guidance, Report GACIAC-PR-80-01, IIT Research Institute, Chicago, Illinois, Nov. 1979, pp. 213-224.

14. Richard A. Steinberg, "Multi-Resolution Spatial Integrator Signal Processor", invention disclosure, JHU/APL File No. 5301-242, 5 March 1986.

15. Dana H. Ballard and Christopher M. Brown, Computer Vision, Prentice Hall, Inc., Englewood Cliffs, NJ (1982).

16. Multiresolution Image Processing and Analysis, A. Rosenfeld, ed., Springer-Verlag, Berlin, 1984.

17. L. O'Gorman and A. C. Sanderson, "A Comparison of Methods and Computation for Multi-Resolution Low- and Band-Pass Transforms for Image Processing", Computer Vision, Graphics, and Image Processing, Vol. 37, pp. 386-401 (1987).

18. Jane's Fighting Ships, 1978-79, Macdonald and Jane's Publishers Limited, London, 1978.

19. The Infrared Handbook, W. L. Wolfe and G. J. Zissis, eds., United States Government Printing Office, Washington, D.C., 1978.

20. P. Perry Ostrowski and Donald M. Wilson, "A Simplified Computer Code for Predicting Ship Infrared Signatures", Report NSWC TR 84-540, Naval Surface Weapons Center, Silver Spring, MD, 13 November 1985.

21. W. McCracken and L. Wajsfelner, "MRTD as a Figure of Merit", in Thermal Imaging, Irving R. Abel, Editor, SPIE Proceedings, Vol. 636, pp. 31-35 (1986).

22. J. D. E. Beynon, "Some Considerations of Signal/Noise Improvement Using Recursive and Nonrecursive Integrators", Electronics Letters, Vol. 14, No. 17, 17 August 1978, pp. 554-555.

23. Herman J. Blinchikoff and Anatol I. Zverov, Filtering in the Time and Frequency Domains, John Wiley & Sons, NY, 1976 (Chapter 8, and refs.).

24. J. Bee Bednar, "On the Approximation of FIR by IIR Digital Filters", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-31, No. 1, Feb. 1983, p. 28.

25. L. M. Howser and W. J. Tropf, "Investigation of Scanning IR Seeker Performance in Background Clutter", JHU/APL F1F(2)86-U-103, July 7, 1986.

26. Richard F. Gasparovic, "Airborne Measurements of Sea Surface Temperature Variability with a Two-Wavelength Infrared Radiometer", presented at the meeting of the American Geophysical Union, June 1982.

27. D. K. White, Jr., "Naval Weapons Center Infrared Image Data Base," JHU/APL F1C(1)83-U-039, June 16, 1983.

28. D. K. White, Jr., "Target Recognition Program Data Base Acquisition", JHU/APL F1C(1)81-U-028, July 23, 1981.

APPENDIX A: FORTRAN LISTING OF MRSI SIMULATION

```
      PROGRAM MRSIPC
C                                         29 APRIL 1987    12:40 PM
      REAL*8        DSEED
      REAL*4        GAIN
      COMMON  /SEED/     DSEED,NSEEDS,ISKIP
      COMMON  /XINIT/    KK1,KK2,K(5),IM(5)
      COMMON  /GAYN/     GAIN(1000)
      DATA    RMS,BCKGND  /1.0,100.0/
C
      CALL RDINPT
C
      NTNRS = 5
      DO 400 ITNR=1,NTNRS
      ITRIAL = 0
      TNR = 0.05 + FLOAT(ITNR) * 0.10
      TGT = BCKGND + TNR * RMS
C
      DO 100 ISEED=1,NSEEDS
      ITRIAL = ITRIAL + 1
      DSEED = DSEED + 1.0D-03
C
C  CREATE NOISE FRAME...
C
      CALL GAUS(RMS)
      CALL ADNOIS(0,1.)
C
C  CALIBRATE MRSI USING NOISE-ONLY FRAME...
C
      ITST = ISKIP * (ISEED / ISKIP)
      IF(ITST.EQ.ISEED) WRITE(22,210)
      CALL INIT(KK2P1)
            DO 200 KK=1,KK2P1
            CALL XAV(KK)
            IMKK = IM(KK)
                 DO 200 I = 1,IMKK
                 CALL IAV(I,KK)
                 CALL BES(I,KK)
                 CALL NE1(I,KK)
 200        CONTINUE
            CALL NE2
C           CALL SIGOUT
C
C  CALIBRATION PROCESS IS COMPLETE.
C  ADD SHIP IMAGE TO NOISE FRAME...
C
            CALL ADSHIP(TGT,BCKGND)
            CALL ADNOIS(1,1.)
C
C  USE MRSI TO LOOK FOR SHIP...
C
            CALL INIT(KK2P1)
            DO 300 KK = 1,KK2P1
            CALL XAV(KK)
            IMKK = IM(KK)
                 DO 300 I = 1,IMKK
                 CALL IAV(I,KK)
                 CALL BES(I,KK)
                 CALL TD(I,KK)
 300        CONTINUE
            CALL TDOUT(TNR,GAIN(ITRIAL),ISEED)
 100  CONTINUE
      IF(ITRIAL.LT.ISKIP) GO TO 400
C
C  GENERATE SUMMARY OF SNR GAIN PERFORMANCE...
C
      CALL GOUT(ITRIAL,ISKIP)
C
 400  CONTINUE
 210  FORMAT(5X,'BEGIN CALIBRATION PROCESS')
      STOP
      END
C
C*********************************************************************
C
      SUBROUTINE RDINPT
C
C** READS INPUT DATA
C
      LOGICAL*1     TEMP(64,256)
      LOGICAL*4     LSHIP(64,256)
      INTEGER*4     ITEMP4(64,256),ISKIP
      REAL*4         ARAY,SHIP
      REAL*8         DSEED
      COMMON /SEED/   DSEED,NSEEDS,ISKIP
      COMMON /XPARAM/ IMAX,KMIN,KMAX
      COMMON /SPARAM/ JMAX,NMAX,JAVG,JDEC
      COMMON /RAWFRM/ ARAY(64,256),SHIP(64,256)
      COMMON /FILTER/ TS,FCO,NORDER,ISET
      EQUIVALENCE (LSHIP,ITEMP4)
C
      READ(21,105)  DSEED,NSEEDS,ISKIP
      WRITE(23,106) DSEED,NSEEDS,ISKIP
 105  FORMAT(1X,D14.7,1X,I3,1X.I2)
 106  FORMAT(1X,'DSEED=',D14.7,3X,'NSEEDS=',I3,3X,'ISKIP=',I2,/)
C
      DO 50 I = 1,64
 50   READ(20,100)  (TEMP(I,J) , J=1,256)
C
      DO 60 J=1,256
      DO 60 I=1,64
      LSHIP(I,J) = TEMP(I,J)
      SHIP(I,J) = ITEMP4(I,J) / 255
 60   CONTINUE
C
      READ(21,110)  IMAX,JMAX,KMIN,KMAX,JAVG,JDEC,NMAX
      WRITE(23,120) IMAX,JMAX,KMIN,KMAX,JAVG,JDEC,NMAX
      READ(21,115)  TS,FCO,NORDER,ISET
      WRITE(23,125) TS,FCO,NORDER,ISET
 100  FORMAT(2(128A1))
 110  FORMAT(2(I3,1X),4(I2,1X),I3)
 115  FORMAT(F9.6,1X,F7.1,1X,I2,1X,I2)
 120  FORMAT(1X,'IMAX=',I3,2X,'JMAX=',I3,2X,'KMIN=',I2,2X,'KMAX=',I2,2X,
     +           'JAVG=',I2,2X,'JDEC=',I2,2X,'NMAX=',I3,/)
 125  FORMAT(1X,'TS=',F9.6,3X,'FCO=',F7.1,3X,'NORDER=',I2,3X,'ISET=',
     +           I2,/)
      RETURN
      END
C
C*********************************************************************
C
```

```
      SUBROUTINE INIT(KK2P1)
C
C** INITIALIZATION FOR ROUTINES XAV & IAV
C
C** 'DO 300' LOOP PERFORMS DETECTION PREPROCESSING
C   (BOXCAR INTEGRATION AND DECIMATION-IN-TIME)
C
C   INPUT (FROM RDINPT):
C             ARAY(I,J)     I=1,2,...,IMAX
C                           J=1,2,...,JMAX
C             IMAX,JMAX
C             KMIN,KMAX     MIN/MAX VALUES (CROSS-SCAN RESOLUTION PARMS.)
C             JAVG          IN-SCAN AVERAGING FACTOR
C             JDEC          IN-SCAN DECIMATION FACTOR
C             NMAX          MAX. VALUE (IN-SCAN RESOLUTION PARAMETER)
C
C   OUTPUT / DATA FRAME WITH IN-SCAN DECIMATION:
C             V(I,J)        J=1,2,...,JMAX2
C
C   OUTPUT / CROSS-SCAN AVERAGING PARAMETERS:
C             KK            INDEX FOR K(KK),IM(KK)
C                          = 1,2,3,...,(KK2+1)
C
C             K(KK)         CROSS-SCAN RESOLUTION PARAMETER
C                          = 1,2,4,8,...,KMAX  (POWERS OF 2)
C
C             IM(KK)        MAX. VALUE OF I FOR V(I,J)
C
C             KK1           SMALLEST VALUE OF KK FOR WHICH CROSS-SCAN
C                           PROCESSING SHOULD BE PERFORMED
C
C             KK2           NO. OF CROSS-SCAN AVERAGING STAGES
C
C   OUTPUT / IN-SCAN AVERAGING PARAMETERS:
C             NN            INDEX FOR N(NN)
C                          =1,2,3,...,(NN2+1)
C
C             NN2           NO. OF PARALLEL CHANNELS OF IN-SCAN PROCESSING
C
C             N(NN)         IN-SCAN RESOLUTION PARAMETER
C                          =1,2,4,8,...,NMAX  (POWERS OF 2)
C
C             JMAX2         MAX. VALUE OF J FOR V(I,J)
C
C-----------------------------------------------------------------------
C   INPUT COMMON BLOCKS  (FROM RDINPT)
              COMMON /XPARAM/ IMAX,KMIN,KMAX
              COMMON /SPARAM/ JMAX,NMAX,JAVG,JDEC
              COMMON /RAWFRM/ ARAY(64,256)
C   OUTPUT COMMON BLOCKS
              COMMON /XINIT/  KK1,KK2,K(5),IM(5)
              COMMON /SINIT/  NN2,N(7),JMAX2
              COMMON /NUFRMS/ V(64,256),VIKK(256,7)
C
      KK1 = INT(1.443*ALOG(FLOAT(KMIN)))
      KK2 = INT(1.443*ALOG(FLOAT(KMAX)))
C
      KK2P1=KK2+1
      DO 100 KK=1,KK2P1
      K(KK) = 2**(KK-1)
      IM(KK) = IMAX - K(KK) + 1
 100  CONTINUE
C
      NN2 = INT(1.443*ALOG(FLOAT(NMAX)))
C
      NN2P1=NN2+1
      DO 200 NN=1,NN2P1
      N(NN) = 2**(NN-1)
 200  CONTINUE
C
      JAVGM1 = JAVG - 1
      JMAX1 = JMAX - JAVGM1
      JMAX2 = JMAX1/JDEC
C
      XJAVG = FLOAT(JAVG)
      JDECM1 = JDEC - 1
C
      DO 300 I=1,IMAX
C
C   PERFORM IN SCAN AVERAGING ...
C
            DO 310 J=1,JMAX1
            SUM = 0.0
            JPJ=J+JAVGM1
                 DO 320 JSUM=J,JPJ
                 SUM = SUM + ARAY(I,JSUM)
 320             CONTINUE
            V(I,J) = SUM/XJAVG
 310        CONTINUE
C
C   PERFORM IN-SCAN DECIMATION ...
C
            JHOP = -JDECM1
            DO 330 J=1,JMAX2
            JHOP = JHOP + JDEC
            V(I,J) = V(I,JHOP)
 330        CONTINUE
 300  CONTINUE
      RETURN
      END
C
C*************************************************************************
C
      SUBROUTINE XAV(KK)
C
C** CROSS-SCAN AVERAGING PROCESSOR, KK-SECTION
C
      COMMON /XINIT/  KK1,KK2,K(5),IM(5)
      COMMON /SINIT/  NN2,N(7),JMAX2
      COMMON /NUFRMS/ V(64,256),VIKK(256,7)
      COMMON /WORK/   TEMP1(64),TEMP2(256,7)
C
      IF(KK.EQ.1) GO TO 100
      IMKK = IM(KK)
      KKM1 = KK - 1
      DO 200 J=1,JMAX2
C
            DO 300 I=1,IMKK
            IPKM1 = I + K(KKM1)
            TEMP1(I) = (V(I,J) + V(IPKM1,J)) / 2.0
 300        CONTINUE
C
            DO 400 I=1,IMKK
            V(I,J) = TEMP1(I)
 400        CONTINUE
 200  CONTINUE
 100  CONTINUE
      RETURN
      END
C
C************************************************************************
C
      SUBROUTINE IAV(I,KK)
C
C** IN-SCAN AVERAGING PROCESSOR / I-K SECTION
C
      COMMON /SINIT/  NN2,N(7),JMAX2
      COMMON /NUFRMS/ V(64,256),VIKK(256,7)
      COMMON /WORK/   TEMP1(64),TEMP2(256,7)
C
      DO 50 J = 1,JMAX2
      VIKK(J,1) = V(I,J)
  50  CONTINUE
C
C   RECURSIVE FILTER INITIALIZATION: VIKK(N(NN),NN)
C
      WNNM1 = V(I,1)
      NN2P1=NN2+1
      DO 100 NN=2,NN2P1
      SUM = 0.0
      J1 = 1 + N(NN-1)
      J2 = N(NN)
C
            DO 200 J=J1,J2
            SUM = SUM + V(I,J)
 200        CONTINUE
C
      WNN = WNNM1 + SUM
      VIKK(J2,NN) = WNN/FLOAT(J2)
      WNNM1 = WNN
 100  CONTINUE
C
C   RECURSIVE FILTER
C
      JM2M1=JMAX2-1
      DO 300 NN=2,NN2P1
      NNN=N(NN)
C
            DO 300 J=NNN,JM2M1
            JP1 = J+1
            JP1MN=JP1-NNN
            SUM = V(I,JP1) - V(I,JP1MN)
            SUM = SUM/FLOAT(N(NN))
            VIKK(JP1,NN) = VIKK(J,NN) + SUM
 300  CONTINUE
C
      RETURN
      END
C
C************************************************************************
```

```
C
      SUBROUTINE ADSHIP(TGT,BCKGND)
C
      COMMON /XPARAM/ IMAX,KMIN,KMAX
      COMMON /SPARAM/ JMAX,NMAX,JAVG,JDEC
      COMMON /RAWFRM/ ARAY(64,256),SHIP(64,256)
C
      DELTGT =   TGT - BCKGND
C
      DO 10 J=1,JMAX
      DO 10 I=1,IMAX
      ARAY(I,J) = BCKGND + DELTGT*SHIP(I,J)
 10   CONTINUE
C
      RETURN
      END
C
C************************************************************************
C
      SUBROUTINE GAUS(SIG)
C
C** CREATES A FRAME OF GAUSSIAN NOISE WITH A WHITE OR DIGITAL
C   BUTTERWORTH POWER SPECTRUM.
C
C   INPUTS REQUIRED:
C          SIG = RMS VALUE OF OUTPUT FRAME
C          DSEED = SEED FOR RANDOM NUMBER GENERATOR
C          ISET = 0    , WHITE NOISE
C                 1    , DIGITAL BUTTERWORTH FILTER (STEARNS)
C
C        IF (ISET .EQ. 1) , ALSO NEED:
C
C          TS = SAMPLING INTERVAL
C          FCO = 3 DB CUT-OFF FREQUENCY
C          NORDER = FILTER ORDER
C
C
C   OUTPUT:
C          GNOISE(64,256) = OUTPUT DATA FRAME
C
C   NOTE:
C     1)  DATA ARE WHITE IN THE FIRST COORDINATE AND BUTTERWORTH-
C          FILTERED IN THE SECOND COORDINATE
C     2)  ISET=1 OPTION UNAVAILABLE ON PC VERSION
C
C     REAL*4   X(16384),Y(16384)
      REAL*8   DSEED
      COMMON   /SEED/     DSEED,NSEEDS,ISKIP
      COMMON   /GSNS/     GNOISE(64,256)
      COMMON   /FILTER/  TS,FCO,NORDER,ISET
      DATA     NR         /16384/
C
C   GENERATE WHITE GAUSSIAN NOISE.
C
CC    CALL GGNML(DSEED,NR,X)
      CALL GGNML$(DSEED,NR)
CC
CC    IF(ISET .NE. 0) GOTO 50
CC
CC    DO 25 K=1,NR
CC    Y(K) = X(K)
CC25  CONTINUE
C     GOTO 60
C
C   APPLY BUTTERWORTH FILTER.
C
C 50  CALL LOBWF(X,Y,NR,TS,FCO,NORDER,5,1)
C
C   CALCULATE STATISTICS OF FILTERED DATA.
C
C60   CALL STAT$$(Y,NR,YMIN,YMAX,YAVE,YVAR,YSIGMA)
      CALL STAT$$(GNOISE,NR,YMIN,YMAX,YAVE,YVAR,YSIGMA,A,B)
C
C   NORMALIZE THE OUTPUT DATA TO ZERO MEAN AND STD.DEV. = 'SIG.'
C
      C = SIG/YSIGMA
      DO 100 J=1,256
      DO 100 I=1,64
C     N = J+(I-1)*256
C     GNOISE(I,J) = C*(    Y(N)    - YAVE)
      GNOISE(I,J) = C*(GNOISE(I,J) - YAVE)
 100  CONTINUE
      RETURN
      END
C
C***********************************************************************
C
      SUBROUTINE ADNOIS(IOPT,TNR)
C
      COMMON  /XPARAM/    IMAX,KMIN,KMAX
      COMMON  /SPARAM/    JMAX,NMAX,JAVG,JDEC
      COMMON  /RAWFRM/    ARAY(64,256)
      COMMON  /GSNS/      GNOISE(64,256)
C
      OPT = FLOAT(IOPT)
      DO 10 J=1,JMAX
      DO 10 I=1,IMAX
      ARAY(I,J) = OPT * ARAY(I,J) + GNOISE(I,J)/TNR
 10   CONTINUE
      RETURN
      END
C
C***********************************************************************
C
      SUBROUTINE NE1(I,KK)
C
C** NOISE ESTIMATOR - ESTABLISHES RMS VALUES OF THE WAVEFORMS OUTPUT
C                         BY 'BES(I,KK)' UNDER CLOSED-COVER CONDITIONS
C
      REAL*4              X(1536)
      COMMON  /XINIT/     KK1,KK2,K(5),IM(5)
      COMMON  /SINIT/     NN2,N(7),JMAX2
      COMMON  /NUFRMS/    V(64,256),VIKK(256,7)
      COMMON  /SIGS/      SIG(5,7),XSUM(64,5,7),XSQR(64,5,7),XNUM(64,5,7)
C
      NN2P1 = NN2+1
      DO 10 NN=1,NN2P1
      INN = 1/NN
      JSTART = 2*N(NN)+INN
      JSTOP = JMAX2-N(NN)-INN
      NI = 0
C
            DO 20 J=JSTART,JSTOP
            NI = NI+1
            X(NI) = VIKK(J,NN)
 20         CONTINUE
C
      CALL STAT$$(X,NI,A,B,C,D,E,XSUM(I,KK,NN),XSQR(I,KK,NN))
      XNUM(I,KK,NN) = FLOAT(NI)
 10   CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE NE2
C
C** NOISE ESTIMATOR (CONTINUATION OF 'NE1')
C
      COMMON  /XINIT/    KK1,KK2,K(5),IM(5)
      COMMON  /SINIT/    NN2,N(7),JMAX2
      COMMON  /SIGS/     SIG(5,7),XSUM(64,5,7),XSQR(64,5,7),XNUM(64,5,7)
C
      KK1P1 = KK1+1
      KK2P1 = KK2+1
      NN2P1 = NN2+1
C
      DO 100 KK = KK1P1,KK2P1
      DO 100 NN = 1,NN2P1
      XAVE        = 0.0
      TSQR        = 0.0
      RTOT        = 0.0
      SIG(KK,NN) = 0.0
      IMKK        = IM(KK)
C
            DO 200 I = 1,IMKK
            XAVE = XAVE + XSUM(I,KK,NN)
            TSQR = TSQR + XSQR(I,KK,NN)
            RTOT = RTOT + XNUM(I,KK,NN)
 200        CONTINUE
C
      IF(RTOT.LE.0.0) GO TO 100
      XAVE = XAVE / RTOT
      ASQR = TSQR / RTOT
      VAR  = ASQR - XAVE * XAVE
      VAR  = VAR * RTOT / (RTOT - 1.0)
      SIG(KK,NN) = SQRT(VAR)
 100  CONTINUE
C
C THIS IS A SHORTCUT, PROVIDING QUICK ANSWERS FOR WHITE NOISE...
C
CC    RT2 = SQRT(2.)
CC    C1 = 0.0
CC    DO 300 NN = 3,NN2P1
CC    C1 = C1 + 1.0
CC    C2 = RT2**C1
CC    DO 300 KK = KK1P1,KK2P1
CC    SIG(KK,NN) = SIG(KK,2)/C2
CC300 CONTINUE
```

```
C
      RETURN
      END
C
C***********************************************************************
C
      SUBROUTINE SIGOUT
C
C** PRINTS OUTPUT OF SUBROUTINE 'NE'.
C
      COMMON  /XINIT/ KK1,KK2,K(5),IM(5)
      COMMON  /SINIT/ NN2,N(7),JMAX2
      COMMON  /SIGS/  SIG(5,7),XSUM(64,5,7),XSQR(64,5,7),XNUM(64,5,7)
C
      KK1P1 = KK1+1
      KK2P1 = KK2+1
      NN2P1 = NN2+1
      WRITE(23,105)
      WRITE(23,110) ((SIG(KK,NN) , NN=1,NN2P1) , KK=KK1P1,KK2P1)
 105  FORMAT(1X,/,1X,'SIGMAS')
 110  FORMAT(6(1X,F7.3))
C
      RETURN
      END
C
C*****************************************************************************
C
      SUBROUTINE BES(I,KK)
C
C** BACKGROUND ESTIMATION & SUBTRACTION
C
      COMMON /XINIT/  KK1,KK2,K(5),IM(5)
      COMMON /SINIT/  NN2,N(7),JMAX2
      COMMON /NUFRMS/ V(64,256),VIKK(256,7)
      COMMON /SIGS/   SIG(5,7),XSUM(64,5,7),XSQR(64,5,7),XNUM(64,5,7)
      COMMON /WORK/   TEMP1(64),TEMP2(256,7)
C
      NN2P1 = NN2+1
C
      DO 200 NN=1,NN2P1
      INN = 1/NN
      JSTART = 2*N(NN)+INN
      JSTOP  = JMAX2-N(NN)-INN
      NNM1 = NN-1+INN
C
            DO 200 J=JSTART,JSTOP
            JL = J-(3*N(NN))/2 - INN
            JR = J+N(NN)+INN
            VHAT = (VIKK(JL,NNM1) + VIKK(JR,NNM1))/2.0
            TEMP2(J,NN) = VIKK(J,NN) - VHAT
 200  CONTINUE
C
      DO 300 NN=1,NN2P1
      INN = 1/NN
      JSTART = 2*N(NN)+INN
      JSTOP  = JMAX2-N(NN)-INN
C
            DO 300 J=JSTART,JSTOP
            VIKK(J,NN) = TEMP2(J,NN)
 300  CONTINUE
```

```
C
      RETURN
      END
C
C******************************************************************************
C
      SUBROUTINE TD(I,KK)
C
C** THRESHOLD / DECISION PROCESSOR
C
      COMMON /XINIT/  KK1,KK2,K(5),IM(5)
      COMMON /SINIT/  NN2,N(7),JMAX2
      COMMON /NUFRMS/ V(64,256),VIKK(256,7)
      COMMON /SIGS/   SIG(5,7),XSUM(64,5,7),XSQR(64,5,7),XNUM(64,5,7)
      COMMON /XCDNS/  JX(64,5),NNX(64,5),VX(64,5)
C
      NN2P1 = NN2+1
C
      DO 100 NN=1,NN2P1
      INN = 1/NN
      JSTART = 2*N(NN)+INN
      JSTOP = JMAX2-N(NN)-INN
C
            DO 100 J=JSTART,JSTOP
            VIKK(J,NN) = VIKK(J,NN)/SIG(KK,NN)
 100  CONTINUE
C
      JX(I,KK) = 1
      NNX(I,KK) = 1
      VX(I,KK) = VIKK(3,1)
C
      DO 400 NN=1,NN2P1
      INN = 1/NN
      JSTART = 2*N(NN)+INN
      JSTOP = JMAX2-N(NN)-INN
C
            DO 400 J=JSTART,JSTOP
            IF(VX(I,KK).GT.VIKK(J,NN)) GOTO 400
            JX(I,KK) = J
            NNX(I,KK) = NN
            VX(I,KK) = VIKK(J,NN)
 400  CONTINUE
C
      RETURN
      END
C
C************************************************************************
C
      SUBROUTINE TDOUT(TNR,GAIN,ISEED)
C
C** PRINTS OUTPUT OF SUBROUTINE 'TD'
C
      COMMON /XCDNS/  JX(64,5),NNX(64,5),VX(64,5)
      COMMON /SPARAM/ JMAX,NMAX,JAVG,JDEC
C
      IF(ISEED.EQ.1) WRITE(23,95)
      IF(ISEED.EQ.1) WRITE(23,100)
      IF(ISEED.EQ.1) WRITE(23,110)
  95  FORMAT(1X,' ',/)
 100  FORMAT(' ',1X,'   TNR       IX     JX     K      N        GAIN')
 110  FORMAT(2X,'  -----      --      --    ---    ---         ---- ')
 210  FORMAT(4X,F5.2,6X,I2,4X,I3,4X,I2,6X,I2,5X,F7.3,/)
C
      IX = 1
      JXO = JX(1,1)
      KKX = 1
      NNXO = NNX(1,1)
      VXO =  VX(1,1)
      DO 300 I = 1,64
      DO 300 KK = 1,5
      IF(VX(I,KK).LE.VXO) GO TO 300
      IX = I
      JXO = JX(I,KK)
      KKX = KK
      NNXO = NNX(I,KK)
      VXO = VX(I,KK)
 300  CONTINUE
C
      J = JDEC*JXO
      K = 2**(KKX-1)
      N = JDEC*2**(NNXO-1)
      GAIN = VXO/TNR
      WRITE(23,210) TNR,IX,J,K,N,GAIN
C
      RETURN
      END
C
C***********************************************************************
C
      SUBROUTINE GOUT(ITRIAL,ISKIP)
C
C** PRINTS OUTPUT SUMMARY OF SNR GAIN PERFORMANCE
C
      REAL*4        GAIN
      COMMON  /GAYN/     GAIN(1000)
C
      WRITE(23,100)
      WRITE(23,200)
      WRITE(23,300)
      WRITE(23,400)
      WRITE(23,500)
 100  FORMAT(1X,' ',//,
     + '   ******************************************************',//)
 200  FORMAT(12X,'SUMMARY OF SNR GAIN PERFORMANCE')
 300  FORMAT(12X,'--------------------------------',//)
 400  FORMAT(' ',4X,'TRIALS',4X,'GMIN',7X,'GMAX',7X,'GAVE',7X,'GSIGMA')
 500  FORMAT(' ',4X,'------',3X,'-------',3(4X,'-------'))
 600  FORMAT(6X,I3,4(4X,F7.3))
C
      DO 700 ITRL = 2,ITRIAL
      ITST = ISKIP * (ITRL / ISKIP)
      IF(ITST.NE.ITRL) GO TO 700
      CALL STATS$(GAIN,ITRL,GMIN,GMAX,GAVE,GVAR,GSIGMA,A,B)
      WRITE(23,600) ITRL,GMIN,GMAX,GAVE,GSIGMA
 700  CONTINUE
C
      WRITE(23,100)
      RETURN
      END
C
```

```
      DO 50 IVEC=1,NR
C
C                 GENERATE NEW PAIR
C            GENERATE FIRST UNIFORM NUMBER
      IF (K.EQ.2) GO TO 40
      K = 2
   10 Z = I * X
      J = IFIX(SNGL(Z))
C     X = Z - DFLOAT(J) + DFLOAT(J)/P
      X = Z -    DBLE(J) +   DBLE(J)/P
      X1 = TWO * X - ONE
      IF(X1.EQ.0) GO TO 10
      IF(X1.GT.ONE) GO TO 10
C            GENERATE SECOND UNIFORM NUMBER
   20 Z = IA * AX
      J = IFIX(SNGL(Z))
C     AX = Z - DFLOAT(J) + DFLOAT(J)/P
      AX = Z -   DBLE(J) +   DBLE(J)/P
      X2 = TWO * AX -ONE
      IF(X2.EQ.0) GO TO 20
      IF(X2.GT.ONE) GO TO 20
C          DETERMINE ACCEPTANCE/REJECTION
      Y1 = X1*X1 + X2*X2
      IF(Y1.EQ.0) GO TO 10
      IF(Y1.LT.ONE) GO TO 30
      GO TO 10
C           TRANSFORMATION TO NORMAL DISTRIBUTION
   30 Y1 = X1 * DSQRT(-TWO * DLOG(Y1)/Y1)
      Y2 = Y1 * (X2/X1)
      XVEC(IVEC) = Y1
      GO TO 50
   40 K = 1
      XVEC(IVEC) = Y2
C
   50 CONTINUE
      RETURN
      END
C
      BLOCK DATA
      INTEGER*2 K
      REAL*8 I,IA,P,Y2
      COMMON/HOLD/I,IA,P,Y2,K
      DATA K/1/,I/4194305.D0/,IA/2097153.D0/
      DATA P/2147483647.D0/,Y2/0.D0/
      END
```